

# 四角形要素の形状と方向性を考慮した 三角メッシュから四角メッシュへの自動変換

伊藤 貴之\* 嶋田 憲司\*\* 井上 恵介\* 古畑 智武\*\*\*

\*日本アイ・ビー・エム(株) 東京基礎研究所

\*\*カーネギーメロン大学 機械工学科 \*\*\*日本アイ・ビー・エム(株) AP ソリューション開発  
{ itot, inoue, furuhata }@trl.ibm.co.jp shimada@cmu.edu

## Automated Conversion of Triangular Mesh to Quadrilateral Mesh with Shape and Directionality Control of Quadrilateral Elements

Takayuki Itoh\* Kenji Shimada\*\* Keisuke Inoue\* Tomotake Furuhashi\*\*\*

\*IBM Research, Tokyo Research Laboratory

\*\*Carnegie Mellon University, Mechanical Engineering \*\*\*IBM Japan, AP Solution Development

**要旨** 本報告では、三角メッシュからの自動変換によって、有限要素解析に適した良質な四角メッシュを自動生成する手法を提案する。本手法では、三角メッシュのメッシュ辺を順に消去し、消去された辺を共有する2個の三角形要素を四角形要素に変換することで、三角メッシュを四角メッシュに返還する。メッシュ辺の消去順は、四角形要素の形状と方向性を考慮して決定される。

### 1. はじめに

力学や電磁気学などの問題を計算機で数値的に求める有限要素解析では、解の精度と収束性の観点において、三角メッシュよりも四角メッシュが好んで用いられることが多い。しかし、三角メッシュと比べると、四角メッシュを自動生成することは難しく、いまま活発な研究が進められている。

ところで、多くの有限要素解析において、四角メッシュには以下のような要件が求められる。

**【要件1】** 複雑な形状においても自動生成できること。

**【要件2】** 四角形要素の大きさを自在に制御できること。重要な部位では小さな四角形要素を、それ以外の部位では粗い四角形要素を生成できることが望ましい。

**【要件3】** 良質な形状の四角形要素が生成されること。ここでいう良質な形状とは「等角に近い」という意味を指す。つまり、正方形や長方形に近い形状の四角形要素を「良質な形状の四角形要素」とする。

**【要件4】** 物理量の移動方向や、領域の境界線の方向など、解析者の意図する方向に多くの四角形要素が整列すること。

従来の自動四角メッシュ生成手法の中には、**【要件4】**を満たす手法はほとんど見当たらない。それに対して筆者らは、上記の要件をすべて満たす自動四角メッシュ生成手法について研究している。その一環として筆者らは、良質な四角メッシュを生成するためのメッシュ頂点生成手法<sup>1)</sup>を提案している。この手法によって発生されたメッシュ頂点を適切に接続できれば、上記の要件を満たす四角メッシュを自動生成することができる。そこで筆者らは、文献1)2)に示す手法によって生成されたメッシュ頂点

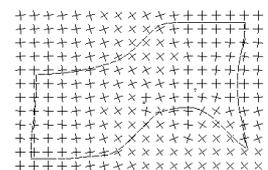
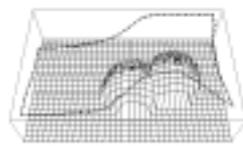
を接続して三角メッシュを生成し、それを四角メッシュに自動変換する手法について研究している<sup>3)</sup>。

本報告では、上記の要件を満たす四角メッシュを自動生成するために、四角形要素の形状と方向性を考慮しながら三角メッシュを四角メッシュに変換する手法を提案する。

### 2. 本報告で提案する四角メッシュ生成手法の概要

本報告では、四角メッシュ生成のために、以下の入力データが与えられることを仮定する(図1(a)(b)参照)。

- 直線または曲線で閉じた形状領域。
- メッシュ要素の大きさを表すスカラ場。



- メッシュ要素の方向性を表すベクタ場。

(a) 形状領域とスカラ場 (b) 形状領域とベクタ場

図1 本報告が仮定する入力データ

筆者らの実装では、この入力データを用いて、まず三角メッシュ(図2(a)参照)を自動生成する。続いて、三角メッシュ中の隣接三角形要素を順にペアにして、その共有辺を消去して四角形要素に変換する。すべての三角形要素をペアにすることは困難なので、少数の三角形要素が残存する三角・四角混合メッシュ

(図2(b) 参照) が生成される。有限要素解析の現場では、このような三角・四角混合メッシュをそのまま使っている事例も多い。この場合、混入する三角形要素の数は、少なければ少ないほど好ましい。

四角形要素だけで構成される四角メッシュを生成するには、三角・四角混合メッシュに含まれる三角形要素および四角形要素を、図 2(c) に示すテンプレートによって分割するのが一般的である。このときも、三角・四角混合メッシュ中の三角形要素の数が少ないほうが、良質な四角メッシュを生成できる。

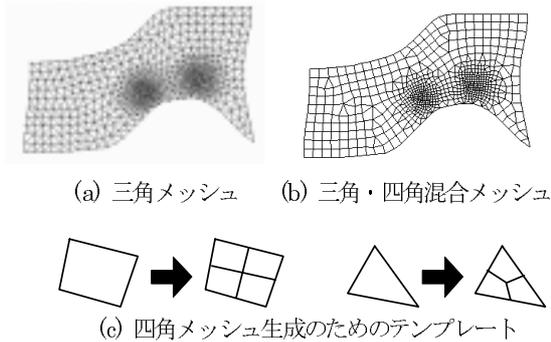


図2 本手法による四角メッシュ生成結果の例

### 3. 四角メッシュ生成のためのメッシュ頂点生成手法

一般的に、メッシュ生成結果はメッシュ頂点の配置に大きく影響される。筆者らは、良質な三角メッシュを生成するためのメッシュ頂点配置手法として、粒子の充填モデルを使った手法<sup>4</sup>を報告している。この手法では、分子間力をもつ円形粒子を領域内部に発生させ、力学シミュレーションによって六方充填する。図 3(a) に示すように、六方充填された円形粒子の中心点を接続することにより、正三角形に近い形状のメッシュ要素を生成することができる。ここで粒子の引力や斥力は、粒子中心点からの距離の関数(図 4(a) 参照)を用いて算出される。

この手法を拡張した手法として、筆者らは、分子間力をもつ四角形粒子を力学シミュレーションによって充填し、その四角形粒子の中心点を接続する(図 3(b) 参照)ことで良質な四角メッシュを生成する、Square Packing 法<sup>2)</sup>を報告している。この手法における四角形粒子の引力や斥力の算出には、四角形粒子の中心点からの距離の関数と、四角形粒子の4頂点からの距離の関数を重ねあわせた、図 4(b) に示すような関数を用いる。四角形粒子の辺の長さは、入力データであるスカラ場(図 1(a)参照)から求められる。四角形粒子の辺の方向は、入力データであるベクタ場(図 1(b) 参照)から求められる。

Square Packing 法の実行例を図 5 に示す。Square Packing 法では、引力・斥力シミュレーションと四角形粒子密度の制御を、反復計算によって行う。その結果、図 5(a) に示す四角形粒子の初期配置の例から、図 5(b) に示すような充填結果を得ることができる。この充填結果では、四角形粒子がスカラ場の示す値に従った密度で、しかもベクタ場に沿った方向に規則的に整列しながら充填されていることがわかる。この四角形粒子の中心点をメッシュ頂点とみなして、それを適切に接続して四角形要素を生成する手法があれば、メッシュ要素の大きさや方向性を自在に制御しな

がら、複雑な形状においても良質な四角メッシュを生成することができる。しかし文献<sup>2)</sup>では、メッシュ頂点を接続する方法については詳しく議論されていない。

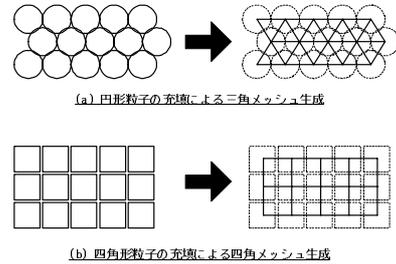


図3 粒子の充填とメッシュ生成

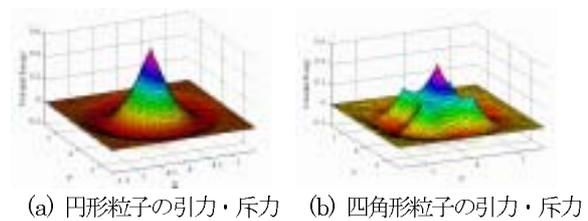


図4 円形粒子と四角形粒子の引力・斥力



図5 四角形粒子の充填結果の例

### 4. 従来のメッシュ変換手法

形状領域内部にメッシュ頂点が生成された後に、それを接続して四角メッシュを生成する手法は、何種類か報告されている。しかしその多くは、実装が複雑である、計算時間がかかる、メッシュ要素の大きさの制御ができない、複雑な形状において四角メッシュの生成に失敗する、というような問題点を残していた。その中で最も有力な手法として、まずメッシュ頂点を接続して三角メッシュを自動生成し、それを四角メッシュに自動変換する、という二段階の処理によって四角メッシュを生成する手法<sup>5(6)7(8)9)</sup>が多く報告されている。三角メッシュ生成ではメッシュ要素の大きさを自在に制御できる方法が多いので、それによって生成された三角メッシュからの変換によって、メッシュ要素の大きさを自在に制御した四角メッシュを生成できる。

メッシュ変換手法によって生成される四角メッシュの特徴は、三角形要素を四角形要素に変換する順番によって決定される。従来手法を三角形要素の変換順番で分類すると、以下の3通りに分類される。

[従来手法 1] 良質な形状の四角形要素を多く生成するように変換順番を決定する手法<sup>6(9)</sup>。

[従来手法 2] 三角メッシュを帯状領域にグループ化することで

四角形要素の規則性を高める手法<sup>8)</sup>。

[従来手法 3] 三角形要素の残存を減らすように変換順番を決定する手法<sup>5)7)</sup>。

しかし、これらの基準を同時に考慮しながら三角メッシュを四角メッシュに変換する手法は見当たらない。

### 5. 四角形要素の形状と方向性、三角形要素の残存可能性を同時に考慮したメッシュ変換手法

本報告では、以下の基準を同時に考慮しながら三角メッシュを三角・四角混合メッシュに変換する手法を提案する。

[基準 1] 良質な形状の四角形要素を優先的に生成する。

[基準 2] 4 辺の方向が角析者の意図する方向に近い四角形要素を優先的に生成する。

[基準 3] 三角形要素を低減するように四角形要素を生成する。

なお、本手法で生成した三角・四角混合メッシュに対して、図 2(c)に示すようなテンプレートを適用することで、四角形要素だけで構成される四角メッシュを生成できる。

本手法では、[基準 1][基準 2][基準 3] を判断するための評価関数を設け、三角メッシュのメッシュ辺に対して関数値を算出し、その関数値の大きい順にメッシュ辺を消去して、そのメッシュ辺を共有する三角形要素を四角形要素に変換する。上記の基準を判断するために本手法では、

[関数 1] 四角形要素の形状を示す関数  $F_a$

[関数 2] 四角形要素の方向性を示す関数  $F_b$

[関数 3] 三角形要素の残存可能性を示す関数  $F_c$

の 3 つの関数を用いる。ここで  $F_a$  および  $F_b$  はメッシュ変換の過程において不変であるので、メッシュ変換を始める前に 5.1 または 5.2 のいずれかの方法で 1 回だけ算出する。 $F_c$  はメッシュ変換の過程において動的に変化するので、5.3 に示す処理手順にしたがって動的に算出される。本手法では、この 3 つの関数の値から算出される 2 種類の関数  $F_1, F_2$  を用いて、メッシュ辺の消去順を決定する。

$$F_1 = aF_a + bF_b, \quad F_2 = F_1 + cF_c$$

ここで、 $a, b, c$  は利用者が定義する定数である。

#### 5.1 メッシュ要素に着目した $F_a$ および $F_b$ の算出

隣接三角形要素  $T_1$  と  $T_2$  をペアにして生成される四角形を  $Q$  とし、その共有辺を  $E$  とする。このとき、 $Q$  の 4 頂点の角度がいずれも 90 度に近づけば、この四角形の形状は良質であると評価される。また、 $Q$  の 4 辺がベクタ場に対して平行または垂直に近づけば、この四角形は角析者の意図する方向に沿っていると評価される。

本手法では、これらの評価を以下のように数式化する。まず  $Q$  の  $k$  番目 ( $k=1,2,3,4$ ) の頂点の角度を  $\alpha_k$  とする。続いて、 $Q$  の中心点における法線方向を  $V_n$  とし、ベクタ場が示すベクタを  $V_a$  とし、 $V_n$  と  $V_a$  の外積ベクタを  $V_b$  とする。 $Q$  の  $k$  番目 ( $k=1,2,3,4$ ) の辺と  $V_a$  または  $V_b$  との角度のうち、小さいほうを  $\beta_k$  とする。このとき  $F_a$  および  $F_b$  の値を、

$$F_a = \Sigma(1.0 \cdot |\cos \alpha_k|), \quad F_b = \Sigma(|\cos \beta_k|)$$

によって算出する (図 6(a) 参照)。この  $F_a$  および  $F_b$  を用いて算出された  $F_1$  の値を  $Q$  の共有辺  $E$  に与える。この値が大き

いほど、 $E$  は優先的に消去され、 $Q$  は優先的に四角形要素に変換される。

なお、この実装方法において、 $a=1.0, b=0.0, c=0.0$  とすれば [従来手法 1] と同等な結果が得られる。同様に、 $a=0.0, b=0.0, c=1.0$  とすれば [従来手法 3] と同等な結果が得られる。

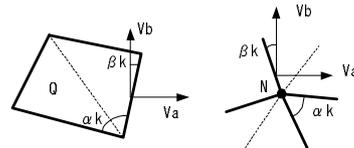
#### 5.2 メッシュ頂点に着目した $F_a$ および $F_b$ の算出

三角メッシュのメッシュ頂点  $N$  に隣接する  $n$  本のメッシュ辺を  $E_k$  ( $k=1, \dots, n$ ) とする。このとき、 $E_k$  の中から  $(n-4)$  本を消去して、頂点  $N$  に隣接する良質な四角形要素を 4 個生成することを考える。

$E_k$  の中から任意の 4 辺を選択し、その 4 辺がつくる角のうちの  $k$  番目 ( $k=1,2,3,4$ ) の角度を  $\alpha_k$  とする。続いて、 $N$  における法線方向を  $V_n$  とし、ベクタ場が示すベクタを  $V_a$  とし、 $V_n$  と  $V_a$  の外積ベクタを  $V_b$  とする。選択された辺のうち  $k$  番目 ( $k=1,2,3,4$ ) の辺と  $V_a$  または  $V_b$  との角度のうち、小さいほうを  $\beta_k$  とする。このとき  $F_a$  および  $F_b$  の値を、

$$F_a = \Sigma(1.0 \cdot |\cos \alpha_k|), \quad F_b = \Sigma(|\cos \beta_k|)$$

によって算出する (図 6(b) 参照)。 $F_a$  の値が大きいくほど、選択された 4 辺は  $N$  の周辺に直角に近い角をつくる。 $F_b$  の値が大きいくほど、選択された 4 辺は角析者の意図する方向に沿っている。このような場合に、選択された 4 辺は良質な四角形要素をつくりやすい。この  $F_a$  および  $F_b$  を用いて算出された  $F_1$  の値を、 $E_k$  の中から辺を 4 本選ぶすべてのパターンに対して同様に行い、最も  $F_1$  の値の大きいパターンを特定する。そのパターンに含まれない  $(n-4)$  本のメッシュ辺に、 $F_1$  の値を与える。大きな値が与えられたメッシュ辺ほど優先的に消去され、それを共有する三角形要素が優先的に四角形要素に変換される。なお、メッシュ頂点が形状領域の境界線上にある場合は、選択する辺の本数を 4 本に限定しない (2 本や 3 本と判断する場合もある)。



(a) メッシュ要素に着目 (b) メッシュ頂点に着目

図 6  $F_a$  および  $F_b$  の算出

#### 5.3 $F_c$ の算出とメッシュ辺の消去順

四角形要素の形状と方向性を考慮しながら、与えられた三角メッシュから適切に多くの三角形要素を四角形要素に変換する解法の一つとして、最大マッチング問題を解くアルゴリズムを適用することが考えられる。三角形要素をノード、三角形要素の共有辺とアークとみなし、 $F_a$  および  $F_b$  の関数  $F_1$  の値でアークを重み付けたグラフに対して、できるだけ多くのノードをマッチングすることで、良質な三角・四角混合メッシュを生成できると予想される。しかし、重みつき無向グラフの最大マッチング問題を解くには、ノード数を  $n$  とすると  $O(n^2)$  を超える計算量を要するといわれている。そこで本報告では、それよりも計算量の少ない実装方法を実現するために、三角形要素の残存可能性を示

す評価関数  $F_c$  を導入する。

三角形要素の残存を減らすことに着目したメッシュ変換手法<sup>2)</sup>では、三角形要素  $T_1$  の隣接三角形要素が  $T_2$  だけになったとき、 $T_1$  と  $T_2$  を優先的に四角形要素に変換する。さもないと、 $T_2$  が  $T_1$  以外の三角形要素とペアになったとき、 $T_1$  が残存することが決定的となってしまふからである。本手法では、この考え方を適用した評価関数  $F_c$  を用いる。ここで、三角形要素  $T_1$  の隣接三角形要素（四角形要素に変換済なものを除く）の数を  $N_i$  とする。 $N_i = 1$  となったとき、 $T_1$  とその隣接三角形要素  $T_2$  の共有辺  $E$  における  $F_c$  の値を  $F_c = 1.0$  とする。それ以外の場合には  $F_c = 0.0$  とする。 $N_i$  の値はメッシュ変換の過程において動的に変化するので、 $F_c$  の値も動的に変化する。

続いて、本手法によるメッシュ辺の消去手順について示す。本手法では最初に、すべてのメッシュ辺に対して  $F_1$  の値を算出し、その値を大きい順にソートしてリスト  $L_1$  に格納する。それと同時に、 $N_i = 1$  である三角形要素を抽出し、それに隣接する三角形要素と共有されるメッシュ辺に対して  $F_2$  の値を算出し、リスト  $L_2$  に格納する。続いて、 $L_1$  の先頭に登録されたメッシュ辺を抽出する。同様に、 $L_2$  に登録されたメッシュ辺のうち  $F_2$  の値が最大であるものを抽出する。前者の  $F_1$  の値  $F_{1max}$  と、後者の  $F_2$  の値  $F_{2max}$  を比較し、大きい値をもつメッシュ辺  $E$  を選択し、 $E$  を  $L_1$  または  $L_2$  から消去する。 $E$  を共有する三角形要素  $T_1$  および  $T_2$  がどちらも四角形要素に変換されていないければ、 $E$  を消去して  $T_1$  と  $T_2$  を四角形要素に変換する。さらに、 $T_1$  および  $T_2$  の最大4個の隣接三角形要素に対して  $N_i$  の値を再算出する。 $N_i = 1$  である三角形要素  $T_p$  があれば、 $T_p$  とその隣接三角形要素  $T_q$  の共有辺に対して、 $F_2$  の値を算出して  $L_2$  に格納する。以上の処理を、 $L_1$  および  $L_2$  が空になるまで反復することで、大半の三角形要素を四角形要素に変換することが出来る。以上の処理の流れを図7に示す。

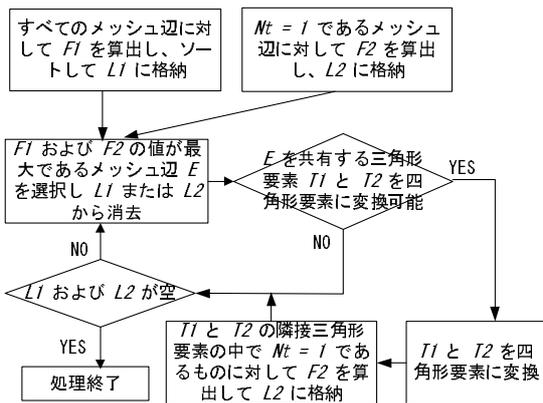


図7 本手法の処理の流れ

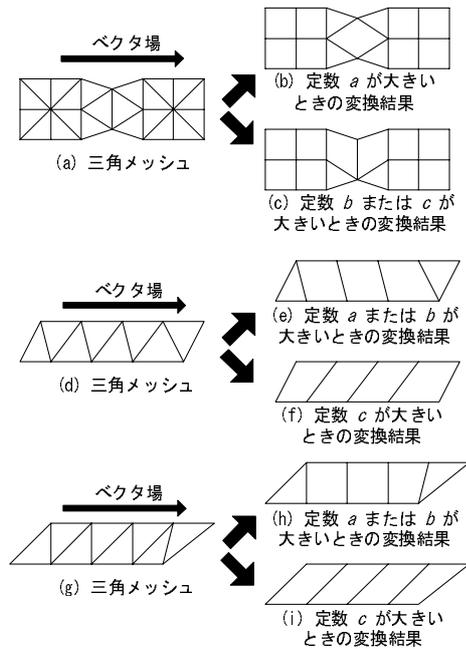
#### 5.4 定数 $a, b, c$ の設定

[基準 1][基準 2][基準 3] の各々の重要度は、有限要素解析手法によって異なるし、解析対象物によっても異なる。定数  $a, b, c$  の値は、その重要度を考慮した上で、実装者というより解析者によって決定されることが望ましい。例えば、Square Packing 法

<sup>2)</sup> のような、四角メッシュのためのメッシュ頂点生成手法を適用したとしても、必ずしも良質な四角形要素だけが生成できるとは限らない。このような場合に、定数  $a, b, c$  の設定は重要な意味をもつ。

例えば図 8(a) のような三角メッシュがあったとき、これを三角・四角混合メッシュに変換すると、定数  $a, b, c$  の値によって、図 8(b)(c) の両方の結果が起こり得る。この結果のどちらが好ましいかは、解析の種類によって異なるので、一概に言えない。逆に言えば、解析者が望ましいと判断するような結果が得られるよう、解析者自身が定数  $a, b, c$  の値を調整することが望ましい。図 8(d)(g) のような三角メッシュにおいても同様に、メッシュ変換結果の善し悪しは一概に言えない。しかも、同相な三角メッシュでありながら、「図 8(d) の三角メッシュは図 8(f) のように変換されて欲しいが、図 8(g) の三角メッシュは図 8(h) のように変換されて欲しい」という複雑な要求も考えられる。本手法では、定数  $a, b, c$  を注意深く調整することで、このような複雑な要求にも応えることが可能となる。これは本手法の大きな特徴である。

図8 定数  $a, b, c$  の設定によるメッシュ変換結果の変化



#### 6. 実行例

本手法を実装してメッシュを変換した結果を示し、その生成結果を数値的に評価する。なお本章では、メッシュの品質が数値的に現れやすいという理由により、四角メッシュではなく三角・四角混合メッシュを比較に用いた。

本章における数値的評価では、三角・四角混合メッシュ中の四角形要素に対して、5.1 で示した評価関数  $F_a$  および  $F_b$  の値を算出し、その平均値 (Average) および最小値 (Worst) を算出した。また、メッシュ全体の要素数に対する四角形要素数の割合 (Quad. ratio) を算出した。

まず、5.1 および 5.2 の 2 種類の実装方法を、2 種類の入力データに対して比較した例を表 1 に示す。この結果から、両者

で非常に類似した結果が得られていることがわかる。他のいくつかの入力データでも試してみたところ、全体的な傾向としては、5.2 に示す実装による結果のほうが、数値の上ではやや良好な結果を示している。

表1 5.1 と 5.2 の実装方法の比較

	データ A		データ B	
	実装 5.1	実装 5.2	実装 5.1	実装 5.2
Average $F_a$	3.5668	3.5684	3.5051	3.5072
Worst $F_a$	1.9364	1.9369	2.0766	1.8099
Average $F_b$	3.9698	3.9711	3.9509	3.9584
Worst $F_b$	3.6491	3.7852	3.1875	3.5088
Quad. ratio	0.9109	0.9109	0.8431	0.8456

続いて、三角メッシュの質の変化に対するメッシュ変換結果の変化を比較した例を表2に示す。Square Packing 法を適用してメッシュ頂点を生成する際には、反復計算の回数が多いほど、メッシュ頂点が適切に配置されるので、メッシュの品質が向上する傾向にある。この比較では、Square Packing 法の反復計算の回数を 5,30,100,1000 回と変化させたときの、三角・四角混合メッシュへの変換結果を比較する。また、このメッシュ変換結果を図9に示す。入力データであるスカラ場には図1(a)に示したものを、ベクタ場には図1(b)に示したものをを用いた。表2より、メッシュ頂点生成時の反復計算回数を大きくするほど、三角・四角混合メッシュの質が向上する傾向にあることがわかる。図9からも、反復回数を大きくするほど、三角形要素が減り、四角形要素の形状や方向性も向上していることが見られる。

表2 三角メッシュの質に対するメッシュ変換結果の比較

	5回	30回	100回	1000回
Average $F_a$	3.3636	3.4079	3.4585	3.5072
Worst $F_a$	1.2994	1.0668	1.0280	1.8099
Average $F_b$	3.8890	3.9228	3.9507	3.9584
Worst $F_b$	3.0326	3.2389	3.3556	3.5088
Quad. ratio	0.8129	0.8234	0.8400	0.8456

続いて、定数  $a, b, c$  の値の変化に対するメッシュ変換結果の変化を比較した例を表3に示す。この比較では、以下の4通りの定数値の組み合わせを用いた。

[定数 A]  $a=0.5, b=1.0, c=0.0$

[定数 B]  $a=1.0, b=0.5, c=0.0$

[定数 C]  $a=0.5, b=1.0, c=1.0$

[定数 D]  $a=1.0, b=0.5, c=1.0$

また、この4通りの定数値によるメッシュ変換結果を図10に示す。入力データであるスカラ場には一定値を用い、ベクタ場には画面の水平方向に一定値のベクタを用いた。表3から、 $a$  を大きくすれば四角形要素の形状がよくなり、 $b$  を大きくすれば四角形要素の方向性がよくなり、 $c$  を大きくすれば四角形要素の数が増加していることがわかる。図10からもこの特徴を確認することができる。例えば、右上の山形部分では、[定数 A] [定数 C] では方向性を重視して、先端の三角形要素の下に水平方向の四角形要素が生成されている。一方 [定数 B] [定数 D] では形状を重視して、先端に正方形に近い形状の四角形要素が生成されている。また、左端の細長い突起部では、[定数 A] [定数 B] では形

状や方向性を重視して、先端に三角形要素を含むメッシュが生成されている。一方、[定数 C] [定数 D] では三角形要素の低減を重視して、突起部にも四角形要素が生成されている。

表3 定数  $a, b, c$  の変化に対するメッシュ変換結果の比較

	定数 A	定数 B	定数 C	定数 D
Average $F_a$	3.3049	3.3575	3.1877	3.3075
Worst $F_a$	1.1716	2.1570	1.1716	1.3421
Average $F_b$	3.8889	3.8709	3.8797	3.8527
Worst $F_b$	2.8284	2.8284	3.4142	2.8284
Quad. ratio	0.6550	0.6453	0.6946	0.6746

## 7. むすび

本報告では、有限要素解析に適した良質な四角メッシュを自動生成するために、四角形要素の形状、方向性、三角形要素の残存可能性、の3つの基準を同時に考慮しながら三角メッシュを四角メッシュに自動変換する手法を提案した。本手法では、3つの基準を判定する評価関数を設定し、三角メッシュのメッシュ辺ごとに評価関数値を算出し、その値の高い順にメッシュ辺を消去する。3つの基準の重要度を調整することにより、幅広くさまざまな種類の解析に適した四角メッシュを自動生成することが可能である。

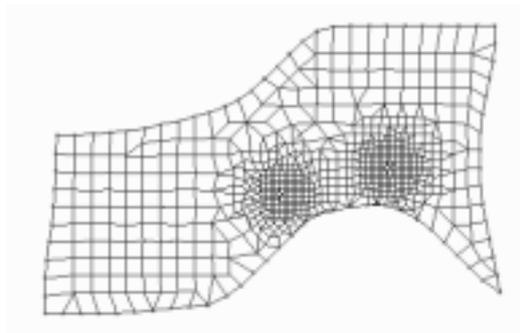
**謝辞** 日頃から活発に議論いただく日本アイ・ビー・エム(株)東京基礎研究所杉本和敏氏、青野雅樹氏、山田教氏、田島玲氏に、感謝の意を表します。

## 参考文献

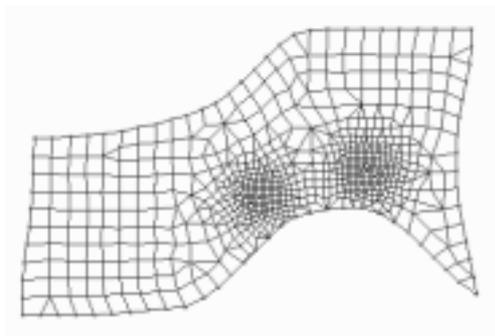
- 1) 伊藤 山田, 井上, 嶋田, 古畑, 平方充填バブル・メッシュ法を用いた自動四角メッシュ生成, 情報処理学会グラフィクスとCAD 研究報告, Vol. 97-CG-87, pp. 7-12 (1997).
- 2) Shimada K., Liao J., and Itoh T., *Quadrilateral Meshing with Directionality Control through the Packing of Square Cells*, Proceedings of 7<sup>th</sup> International Meshing Roundtable, pp. 61-75 (1998).
- 3) Itoh T., Shimada K., Inoue K., Yamada A., and Furuhashi T., *Automated Conversion of 2D Triangular Mesh into Quadrilateral Mesh with Directionality Control*, Proceedings of 7<sup>th</sup> International Meshing Roundtable, pp. 77-86, 1998.
- 4) 嶋田, 物理モデルによる自動メッシュ分割, シミュレーション, Vol. 12, No. 1, pp. 11-20 (1993).
- 5) Heighway E. A., *A Mesh Generator for Automatically Subdividing Irregular Polygons into Quadrilaterals*, IEEE Transactions on Magnetics, Vol. Mag-19, pp. 2535-2538 (1983).
- 6) Lo S. H., *Generating Quadrilateral Elements on Plane and over Curved Surfaces*, Computer and Structures, Vol. 31, No. 3, pp. 421-426 (1989).
- 7) Johnston B. P., Sullivan J. M., and Kwasnik A., *Automatic Conversion of Triangular Finite Element Meshes to Quadrilateral Elements*, International Journal for Numerical Methods in Engineering, Vol. 31, pp. 67-84 (1991).
- 8) Shimada K., and Itoh T., *Automated Conversion of 2D Triangular Mesh into Quadrilateral Mesh*, Proceedings of

International Conference on Computational Engineering Science '95, pp. 350-355 (1995).

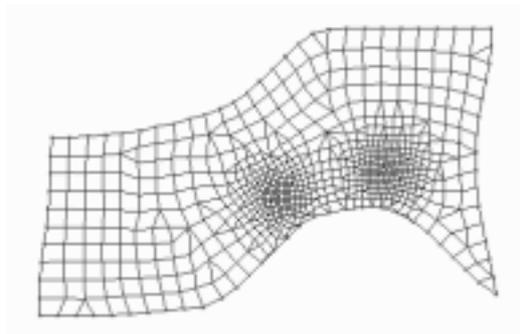
9) Borouchaki H., Frey P. J., and George P. L., *Unstructured Triangle-Quadrilateral Mesh Generation. Application to Surface Meshing*, Proceedings of 5<sup>th</sup> International Meshing Roundtable pp. 229-242 (1996).



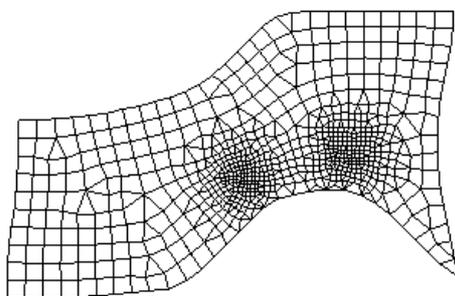
(a) 反復回数 5回



(b) 反復回数 30回

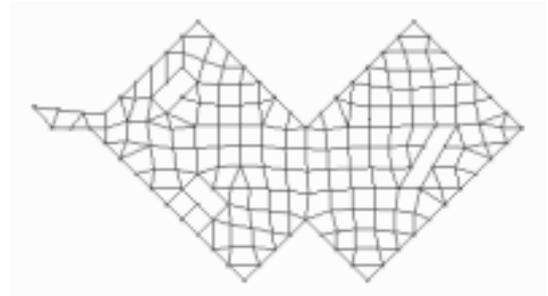


(c) 反復回数 100回

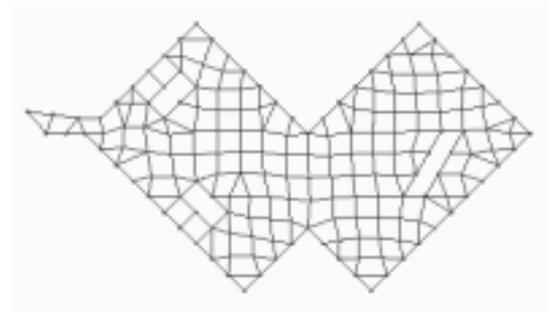


(d) 反復回数 1000回

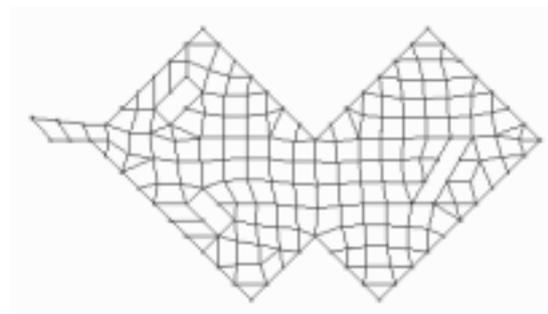
図9 反復回数の変化とメッシュ変換結果



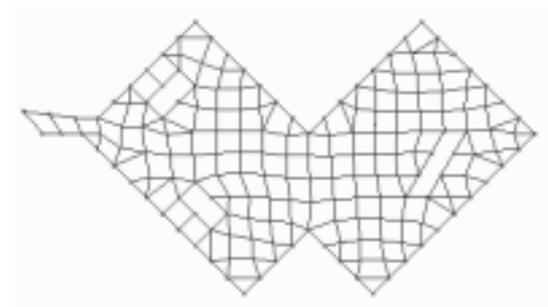
(a) 定数 A



(b) 定数 B



(c) 定数 C



(d) 定数 D

図10 定数  $a, b, c$  の変化とメッシュ変換結果