グラフィクスとCAD 80－1
（1996. 5. 14）

バブル・メッシュ分割法のための要素サイズの制御方法

山田 敦　　嶋田 憲司　　伊藤 貴之
{ayamada, kenjis, itot}@trl.ibm.co.jp
日本アイ・ビー・エム(株) 東京基礎研究所

概要 本報告では，指定された要素サイズを満たしながら要素サイズが連続的に変化するように，メッシュの要素サイズを制御する手法を提案する．領域のどの部分にどれくらいの大きさの要素を生成するかを領域全体にわたって規定した関数は要素サイズの分布関数と呼ばれ，本手法では，この分布関数を薄板の曲げモデルの歪みエネルギーを最小化する手法を使って生成する．一般に要素サイズが急激に変化する場所では極端に歪んだ要素が発生し，有限要素解析結果などに悪影響を与える．本手法を使うと，要素サイズが特に指定されていない領域においても要素サイズが連続的に変化するため，指定された場所での要素サイズを守りながら領域全体にわたって質の良いメッシュを生成することが可能となる．
キーワード CAE, メッシュ分割，有限要素法

Element Size Control for Bubble Mesh Generation Method

Atsushi Yamada,　　Kenji Shimada,　　Takayuki Itoh
{ayamada, kenjis, itot}@trl.ibm.co.jp
IBM Research, Tokyo Research Laboratory

Abstruct We propose a new method of generating a mesh which element size gradually varies while satisfying the constraints given by user. This method generates a smooth node spacing function defined over the given geometric domain by using minimization of the distortion energy of a thin plate bending model. In the mesh area where element size varies greatly, badly distorted elements are generated. Our method generates well-shaped and well-graded mesh, because node spacing function varies smoothly.
Key Words CAE, Mesh Generation, FEM

1 はじめに

与えられた任意形状の二次元領域を要素(三角形や四角形)の集合に分割する技術は自動メッシュ生成と呼ばれ，CAE分野における有限要素解析やCG分野におけるラジオシティー法などをはじめとし，CAE, CG, CADの分野で広く利用される技術である．自動メッシュ生成時の要求事項として以下のものが挙げられる．

- 任意形状の領域に適用できること．
- 領域内の任意の場所で要素サイズが指定できること．すなわち与えられた領域の中で重要な部分ではその重要度に応じた細かさの要素を生成し，そうでない部分には，粗い要素を生成したいという要求である．
- 生成される要素の歪みが少ないこと．これは可能な限り正三角形や長方形に近い要素の生成を意味する．極端に歪んだ要素は，このメッシュを使った有限要素解析結果などに悪影響を与える．
- メッシュの要素サイズが連続的に変化すること．もし要素サイズが急激に変化すると，その場所に極端に歪んだ要素が生成される．

以上の要求を全て満たす方法として,「バブル・メッシュ法」がすでに提案され[2], [5]，応用分野への適用が始まっている．バブル・メッシュ法では，入力として「領域形状」と適切な「要素サイズの分布関数」が与えられると，その後は全自動で上記の要求を満たすメッシュを生成することが出来る．

ここでいう要素サイズの分布関数とは，領域のどの部分にどれくらいの大きさの要素を生成するかを領域全体にわたって規定する関数のことであるが，この要素サイズの分布関数を適切に与えることが出来なければ，上記のメッシュ生成に対する要求の2番目以降が満たされなくなる．したがって，以下の4点の要件を満たす分布関数を生成することが，バブル・メッシュ法の特徴を最大限に引き出すために重要となる．

- **要件 1**: 任意形状の二次元領域に対して分布関数が生成できること．
- **要件 2**: 領域上の任意の場所で指定された要素サイズを満たす分布関数であること．
- **要件 3**: 要素サイズが急激に変化せず，連続的に変化する分布関数であること．
- **要件 4**: 少ない入力データで生成できる分布関数であること．

本報告では，上記4つの要件を満たす要素サイズ分布関数を自動的に生成するという問題に対して，薄板の曲げモデルの歪みエネルギーを最小化することによって，この要素サイズ分布関数を生成する手法を提案する．

2 従来技術

上記4つの要件を全て満たす要素サイズの分布関数を生成することは，一般に容易ではない．従来から比較的よく用いられている分布関数生成技術及びその問題点についてまとめる．

- **従来技術 1**: 分布関数を単一の関数式により表現する方法である．領域形状が単純であり，指定される要素サイズの位置や数が限られている場合には，この方法で分布関数を求めることは可能である．例えば要素サイズがある一定の方向に向かって徐々に増加していくだけといった単純な制御しか要求されない場合には，この従来技術は有効である．しかし任意形状の領域内に任意の位置に任意数の要素サイズが指定された場合，これらの要求を満たす単一関数式を見つけることはきわめて困難である．
- **従来技術 2**: 与えられた領域をいくつかの部分領域に分割し，各部分領域ごとに一定の要素サイズを指定する方法である(図1) [1]．この方法は，部分領域間で要素サイズが連続的に変化せず離散的に変化するため，各部分領域に対してうまく要素サイズを指定しないと，急激な要素サイズの変化のために歪んだ要素が生成される可能性がある．また特定の場所にだけ要素サイズを指定したい場合でも，全ての部分領域に対して要素サイズを入力する必要があるため，入力データが多くなる．

- **従来技術 3**: 領域全体にわたって一定値をとる分布関数に，部分的に指定した分布関数をブレンドし，小さい方の関数値を要素サイズとして採用するという方法である (図 2)[6]．部分的な分布関数が指定されていない部分では，一定の要素サイズをとらざるをえないという問題がある．また部分的に指定した分布関数どうしが重なる場合，その重なった部分で要素サイズが急激に変化する可能性がある．

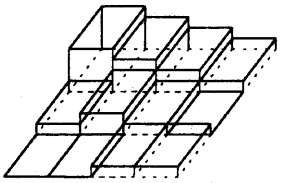


図 1: 従来技術 2

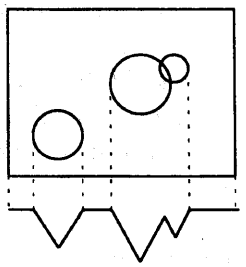


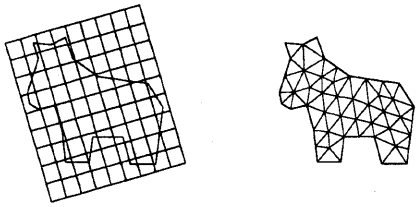
図 2: 従来技術 3

3 要素サイズ分布関数の生成

二次元領域と，その領域内の任意の点に指定された要素サイズとが入力データとして与えられる．この入力データを基に以下の手順で要素サイズの分布関数を生成する．

3.1 領域を覆う平面形状及びその内部の要素の生成

与えられた領域を覆う平面形状を生成する．この平面形状は，図 3に示すような直交格子形状，あるいは入力領域の形状そのもののどちらかであるとする．直交格子形状で覆う場合は，その内部を四角形要素に分割し，入力領域形状そのもので覆う場合は，その内部をデローニ三角形分割法を使って三角形要素に分割する (図 3)．入力形状そのもので覆う場合には，要素サイズが指定された点に三角形の頂点が生成されるように三角形分割を行なう．どれくらいの細かさの要素に平面形状を分割するかは，処理の精度と処理時間との兼ね合いにより決定する．



(a) 四角形要素　　(b) 三角形要素

図 3: 与えられた領域を覆う平面形状およびその要素への分割

3.2 指定された要素サイズの要素頂点への割り付け

入力データとして指定された要素サイズを，その指定された点の近傍にある要素の頂点に割り付ける．

三角形要素を用いる場合には，3.1節で述べたように既に要素サイズが指定された点に三角形の頂点が生成されているので，その点に指定された要素サイズを割り付ければよい．

四角形要素を用いる場合には，要素サイズが指定された点に四角形の頂点があるとは限らないので，要素サイズが指定された点の近傍にある四角形頂点に値を割り付ける (図 4)．いま要素の頂点 V_a に割り付けられる値 f_a を求めることを考える．頂点 V_a に隣接している要素内に指定されている要素サイズを f_i $(i = 1, 2, ...)$，指定されている点を V_i $(i = 1, 2, ...)$ とする．頂点 V_a に隣接している要素内に要素サイズが指定されていれば，以下の式を使って割り付けられる値 f_a を求める．指定されていなければ，その

頂点には値を割り付けない．

$$f_a = \frac{\sum_i w_i f_i}{\sum_i w_i}, \tag{1}$$

$$w_i = 1/l_i, \tag{2}$$

$$l_i = \|V_a - V_i\|. \tag{3}$$

ただし $\|V_a - V_i\|$ は 2 点 V_a, V_i 間の距離を表す．この割り付けは，要素サイズを指定された点が，要素頂点に近いほど強い影響を与えるといったヒューリスティックが用いられている．

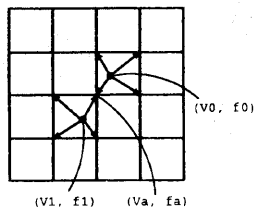


図 4: 指定された要素サイズの要素頂点への割り付け

3.3 歪みエネルギー最小化手法による分布関数の生成

まず生成した個々の要素の内部を形状関数 $z(x,y)$ を使って表現する．四角形要素を用いる場合は，図 5(a) のように一つの四角形要素に対して局所座標 xyz をとり，この要素に対する形状関数 $z(x,y)$ を次のように定める．

$$\begin{aligned} z(x,y) &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy \\ &+ \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 x^2 y + \alpha_9 xy^2 \\ &+ \alpha_{10} y^3 + \alpha_{11} x^3 y + \alpha_{12} xy^3. \end{aligned} \tag{4}$$

係数 α_1 から α_{12} は，四角形要素の 4 頂点での変位 z 及び x,y 方向の回転 θ_x, θ_y が定まれば，一意に決定される．一方，三角形要素を用いる場合は，図 5(b) のように一つの三角形要素に対して局所座標 xyz 及び面積座標 $L_1L_2L_3$ をとり，この要素に対する形状関数を次のように定める．

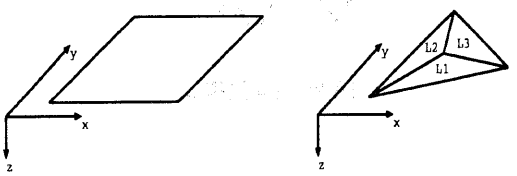
$$\begin{aligned} z(x,y) &= \beta_1 L_1 + \beta_2 L_2 + \beta_3 L_3 \\ &+ \beta_4 (L_1^2 L_2 + L_1 L_2 L_3/2) \\ &+ \beta_5 (L_2^2 L_3 + L_1 L_2 L_3/2) \\ &+ \beta_6 (L_3^2 L_1 + L_1 L_2 L_3/2) \\ &+ \beta_7 (L_1 L_2^2 + L_1 L_2 L_3/2) \\ &+ \beta_8 (L_2 L_3^2 + L_1 L_2 L_3/2) \\ &+ \beta_9 (L_3 L_1^2 + L_1 L_2 L_3/2). \end{aligned} \tag{5}$$

係数 α_1 から α_9 は，三角形要素の 3 頂点での変位 z 及び x,y 方向の回転 θ_x, θ_y が定まれば，一意に決定される．ここで述べた 2 種類の形状関数は，薄板の曲げ問題で最も標準的に利用される形状関数である [4].

このように定めた形状関数に対して，領域を覆う形状の歪みエネルギー U は，次のように定義される．

$$U = \frac{h}{2} \int_S \left(\frac{\partial^4 z}{\partial x^4} + 2 \frac{\partial^4 z}{\partial x^2 \partial y^2} + \frac{\partial^4 z}{\partial y^4} \right) dxdy. \tag{6}$$

ただし h は要素の厚みを表し，$\int_S$ は領域全体にわたる積分を表している．3.2節で要素頂点へ割り付けた値を境界条件として，歪みエネルギー U を最小にするように，各要素頂点の z 方向の変位及び x,y 方向の回転 θ_x, θ_y を有限要素法を使って計算する [4]. z, θ_x, θ_y により各要素の形状関数 (式 (4) あるいは式 (5)) が決定され，この形状関数を要素サイズの分布関数とする (図 6).



(a) 四角形要素　　(b) 三角形要素

図 5: 局所座標における四角形要素及び三角形要素

4 実行結果

3章の方法を使って生成した要素サイズの分布関数を，四角形要素を用いた場合を例に示す．図 7(a) は入力形状を，図 7(b) は入力データとして指定された要素サイズを表す．図 7(b) において，入力形状から

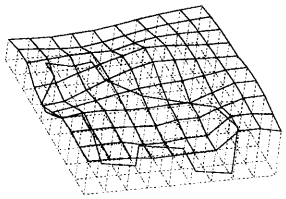


図 6: 四角形要素を使った要素サイズ分布関数

垂直方向に伸びた線の長さが，その点での要素サイズを表している．指定された要素サイズから本方法を使って生成した要素サイズの分布関数を図 7(c) に示す．指定された点の間をなめらかに補間する分布関数が生成されている様子がわかる．この例において分布関数を生成するのに要した時間を PowerPC 603e を搭載した PC で測定した結果，約 1 秒程度であった．

この分布関数の値をバブルの半径として，バブル・メッシュ法により領域にバブルを細密充填した様子を図 7(d) に示す．最後にバブルの中心をデローニ三角形分割法で結んでできた三角形メッシュを図 7(d) に示す．質のよい三角形要素が，指定された要素サイズにしたがって生成されている様子がわかる．またこの三角形メッシュを文献 [3] の手法を使って四角形メッシュに変換することができる．

図 8は，入力形状内に穴及び稜線がある例である．前の例と同様に，図 8(c) でなめらかに変化する分布関数が生成され，図 8(e) で質のよい三角形要素が，指定された要素サイズにしたがって生成されている様子がわかる．

5 おわりに

本報告では，要素サイズの分布関数を，薄板の曲げモデルの歪みエネルギーを最小とすることによって生成する手法について述べた．本手法を用いることにより，要素サイズが特に指定されていない領域においても要素サイズが連続的に変化するため，指定された場所での要素サイズを守りながら領域全体にわたって質の良いメッシュを生成することができた．

本手法で生成したメッシュは，要素サイズに関してはユーザーの意図を反映したものになっているが，メッシュパターンに関してはユーザーの意図を反映しているわけではない．解析に重要な部分では，特定のメッシュパターンが要求されることがある．今後，ユーザーの意図を反映したメッシュパタンを重要な部分に組み込むことを検討していく予定である．

参考文献

[1] J. C. Cavendish, D. A. Field, W. H. Frey, An Approach to Automatic Three-Dimensional Finite Element Mesh Generation, International Journal for Numerical Methods in Engineering. Vol. 21, pp. 329-34, 1985.

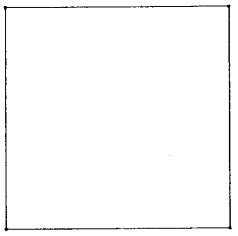
[2] K. Shimada, Physically-Based Mesh Generation: Automated Triangulation of Surfaces and Volumes via Bubble Packing, Ph.D. thesis Massachusetts Institute of Technology, Cambridge, MA, 1993.

[3] K. Shimada, T. Itoh, Automated Conversion of 2D Triangular Mesh into Quadrilateral Mesh, International Conference on Computational Engineering Science '95 Proceedings, pp. 350-355, 1995.

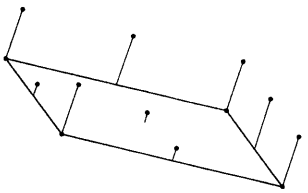
[4] O. C. Zienkiewicz, The Finite Element Method Third Edition, McGraw-Hill, United Kingdom, 1977.

[5] 嶋田憲司, 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. 11-19, 1993.

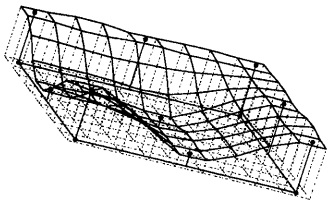
[6] 矢川元基, 吉村忍, 計算力学のためのエキスパートシステム, 応用数理, Vol. 2, No. 2, pp. 30-55, 1992.



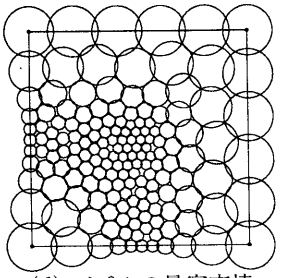
(a) 入力形状



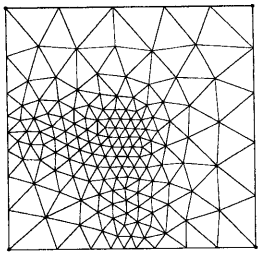
(b) 入力要素サイズ



(c) 要素サイズ分布関数

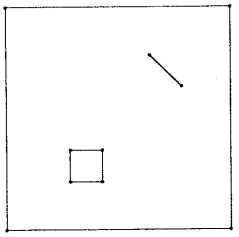


(d) バブルの最密充填

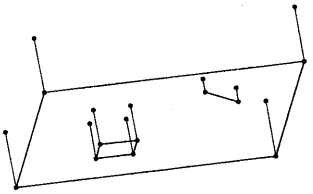


(e) 三角形メッシュ

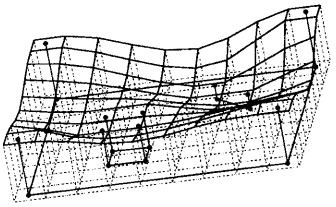
図 7: 分布関数の生成と三角形メッシュ生成例 1



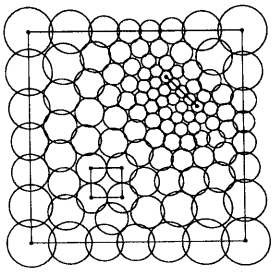
(a) 入力形状



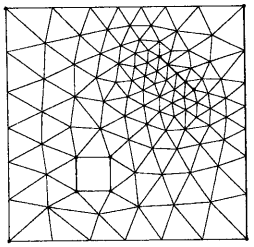
(b) 入力要素サイズ



(c) 要素サイズ分布関数



(d) バブルの最密充填



(e) 三角形メッシュ

図 8: 分布関数の生成と三角形メッシュ生成例 2