

制約つき Delaunay 三角メッシュ生成法の効率的な実装方法

3AD-1

伊藤貴之* 山田敦* 井上恵介* 古畑智武** 嶋田憲司***

* 日本アイ・ピー・エム株式会社 東京基礎研究所

** 日本アイ・ピー・エム株式会社 AP ソリューション開発

*** カーネギー・メロン大学 機械工学科

{itot, ayamada, inoue} @trl.ibm.co.jp, furuhata@yamato.ibm.co.jp, shimada@cmu.edu

1 はじめに

CADなどで生成された形状を細かい要素の集合に分割するメッシュ生成技術は、計算力学、形状表現、画像生成などの各分野に広く用いられている。もっとも有名なメッシュ生成手法の一つに、Delaunay 三角メッシュ生成法がある。この方法は、平面中に与えられた多数の節点(ノード)を連結して三角形要素の集合を生成する方法であり、三角形要素の最小角度が最大になるようにメッシュを生成する。特に、形状の外周(Outside loop)、穴(Inside hole)、内部線分(Inside wire)などを制約条件にして(図1(a)参照)、制約条件を破損しないように三角メッシュを生成する、制約つき Delaunay 三角メッシュ生成法が広く実用されている。ここでいう「制約条件を破損しない」とは、制約条件を構成するすべての線分(制約線分)が1個または2個の要素の辺となるように三角メッシュを生成することを意味する。

制約つき Delaunay 三角メッシュ生成法の典型的なアルゴリズム[1]では、制約線分がない状態で三角メッシュを生成し、そのメッシュに制約線分を追加し、制約線分と交差する要素を除去し、その領域におけるメッシュを再生成する(図1(b)参照)。しかしこの手法では、非常に多数のノードや制約線分をもつ場合に、要素と制約線分の交差判定処理、および再メッシュ生成処理の処理時間が大きくなる。

本報告では、要素と制約線分との交差判定などの処理時間を低減する、制約つき Delaunay 三角メッシュ生成法の効率的な実装方法(図1(c)参照)を提案する。本手法では、まず制約線分に接するノードを処理し、2辺が同一制約線分と交差する三角形がある場合には、メッシュをそのノードの処理前の状態に戻し、別のノードを先に処理する。この処理を終えた時点で、すべての制約線分は1個または2個の三角形要素に共有された状態となる。その後、制約線分に接しないノードを処理して三角メッシュを完成する。この時、制約線分に接しないノードの処理において、新しい三角形要素と制約線分の交差判定をすることなく、制約線分の破損を防ぐことができる。よって、従来の手法よりも、制約線分との交差判定を少なくすることができる。

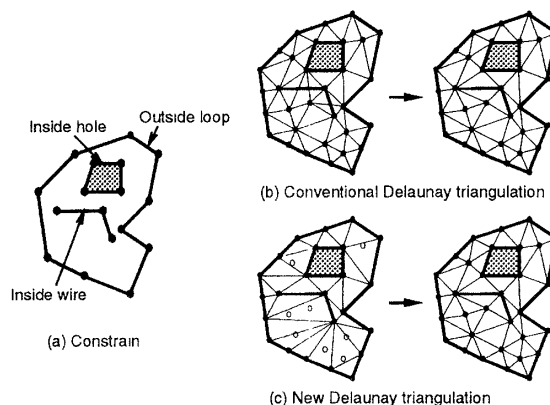


図1: 制約つき Delaunay 三角メッシュ生成法

2 制約つき Delaunay 三角メッシュ生成法の従来の実装方法

文献[1]に示されている制約つき Delaunay 三角メッシュ生成法のアルゴリズムでは、領域を覆う大きな三角形(Super-triangle)を再帰的に分割する以下のような処理手順を用いている。

1. 領域を覆う大きな三角形(Super-triangle)を生成する。
2. ノードを1個ずつ選択し、以下の処理を行う。
 - ノード D を内部に含む三角形 ABC を抽出する。ノード D が1個目のノードである場合、Super-triangle そのものが抽出される。
 - 三角形 ABC を、3つの三角形 ABD, BCD, CAD に分割する。
 - ノード D に接するそれぞれの三角形について、D の対辺を共有する隣接三角形がある場合には、その隣接三角形と以下の変換処理1を行う。この処理は、ノード D に接するすべての三角形について、変換が不要になるまで反復される。
3. すべてのノードについて処理が終了した後に、制約線分と三角形要素の交差判定をする。交差している要素を消去し、要素が消去された領域においてメッシュを再生成する。
4. 領域の外側に生成された三角形を除去し、メッシュを完成する

ここで、変換処理1は、以下のような処理をさす。

変換処理1: 三角形 ABD と三角形 AEB について、ABD の外接円の内部に E がある場合、あるいは AEB の

外接円の内部に D がある場合には、辺 AB を消去して辺 DE を生成することにより、ABD と AEB を三角形 AED と BDE に変換する。さもなければ、2 三角形は変換されない。

3 本手法による制約つき Delaunay 三角メッシュ生成法

本報告で提案する制約つき Delaunay 三角メッシュ生成法の処理手順を、以下に示す。

1. 領域を覆う大きな三角形 (Super-triangle) を生成する。
 2. 制約線分に接するノードを 1 個ずつ選択し、以下の処理を行う。
 - 従来手法と同様に、ノード D を内部に含む三角形 ABC を ABD, BCD, CAD に分割する。
 - ノード D に接するすべての三角形に対して、以下の**変換処理 2**を行う。この結果「変換される」とも「変換されない」とも判定されなかった場合に限り、**変換処理 1**を実行する。この処理は、ノード D に接するすべての三角形が変換されなくなるまで反復する。
 - ノード D に接するすべての三角形に対して、要素辺と制約線分との幾何的な交差判定を行う。
 - 2 辺が同一制約線分と交差している三角形が抽出された場合には、ノード D の処理前の状態にメッシュをもどし、ノード D 以外のノードを選択する。
 3. 制約線分に接するノード以外のノードを 1 個ずつ選択し、以下の処理を行う。
 - 従来手法と同様に、ノード D を内部に含む三角形 ABC を ABD, BCD, CAD に分割する。
 - ノード D に接するすべての三角形に対して、以下の**変換処理 2**を行う。この結果「変換される」とも「変換されない」とも判定されなかった場合に限り、**変換処理 1**を実行する。この処理は、ノード D に接するすべての三角形が変換されなくなるまで反復する。
 4. すべてのノードについて処理が終了した後で、領域の外側に生成された三角形を除去し、メッシュを完成する。
- ここで、**変換処理 2** は、以下のような処理をさす。

変換処理 2：三角形 ABD と三角形 AEB について、辺 DE が制約線分である場合には、ABD と AEB を三角形 AED と BDE に変換する。逆に辺 AB が制約線分である場合には、2 三角形は変換してはいけない。

本手法では、制約線分に接するノードの処理において、制約線分に 2 辺が交差する三角形要素を生成しないように処理を進める。図 2 は、その処理の過程を示したものである。ここで、図 2 (1) は入力形状の制約線分を示したものである。この制約線分に接するノードを、例えば図 2 (2) に示す順番で処理すると、三角形 2-3-4 は 2 辺 2-3 および 4-2 が同一制約線分と交差する。本手法では、このような三角形を生成しないために、図 2 (2) におけるノード 4 の処理を後回しにして、その次の

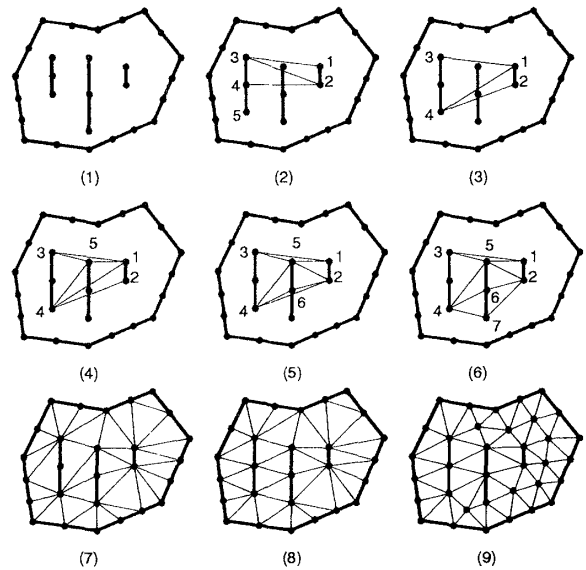


図 2: 本手法による三角メッシュ生成

ノードを先に処理し、図 2 (3) のように三角形を生成する。

また、制約線分に接するノードの処理において、1 辺だけが制約線分に交差する三角形要素が生じることもあるが、これは **変換処理 2** によって確実に解消される。図 2 (3) では、三角形 1-3-4 および 1-4-2 は、2 辺が同一制約線分に交差することはないものの、1 辺だけが同一制約線分に交差している。これらの三角形の交差は、その後の処理（図 2 (4)(5)(6) 参照）で確実に解消される。

その結果、後回しにしたノードを含めて、制約線分に接するノードの処理を終えた時点（図 2 (8) 参照）で、制約線分との交差はなくなり、すべての制約線分は 1 個または 2 個の三角形要素に共有される。よって、それ以降の、制約線分に接しないノードの処理（図 2 (9) 参照）では、すべての制約線分に **変換処理 2** が適用され、制約線分を共有する三角形要素の辺は変換されない。よって、制約線分は破損されることなくメッシュが生成されるので、この過程における要素と制約線分との交差判定は不要である。

従来の手法では、すべてのノードを処理して多数の三角形を生成してから制約線分との交差判定を行っていた。それに対して本手法では、制約線分に接するノードのみで生成された少数の三角形に対して制約線分との交差判定を行うので、従来の手法よりも効率的である。

参考文献

- [1] Sloan S. W., *A Fast Algorithm for Generation of Constrained Delaunay Triangulations*, *Computers and Structures*, Vol. 47, No. 3, pp. 441-450, 1993.