

要素分割に関する技術ノート

徳永 健一*

Notes on Subdivision of Elements

Ken-ichi Tokunaga*

REVOCAP_Refiner による解析モデルの細分方法およびその他の要素分割に関する話題を取り上げる。

Key word: refine element, subdivision, tetrahedron, hexahedron

1. REVOCAP_Refiner による要素分割

REVOCAP_Refiner は文部科学省次世代 IT 基盤構築のための研究開発「イノベーション基盤シミュレーションソフトウェアの研究開発」における「次世代ものづくりシミュレーションソフトウェア作成プロジェクト」で開発された、大規模アセンブリモデルの連成解析に対応した解析モデルの細分化ツールである。

REVOCAP_Refiner は、マルチ力学シミュレーションソフトウェアの一部として、マルチ力学シミュレーションソフトウェアで開発されるソルバー、カップラに組み込まれて使われ、オンメモリ上で有限要素法、有限体積法の非構造格子の要素を細分するライブラリとして利用される。

メッシュ生成、境界条件設定、領域分割等の通常のプレ処理が行われた解析モデルについて、ソルバーまたはカップラはオンメモリで解析モデルを細分することができる。

単純に細分するだけでは、曲面形状の幾何的な解像度は細分前のメッシュの解像度を越えることができない。REVOCAP_Refiner では、この点にも留意し、要素細分のときに、メッシュ生成に用いた CAD モデルを再度用いて、要素細分と同時に形状の補正を行い、細分された解析モデルの CAD モデルに対する形状の解像度を改善する。

1.1. 4 面体 1 次要素の細分

以下の手順で行われる。

- ① それぞれの辺に中間節点を追加する
- ② 元の 4 面体の頂点を含むような 4 つの 4 面体と中央の 8 面体に分割する
- ③ 中央の 8 面体について最も長さの短い対角線を追加して、4 つの 4 面体に分割する

結果として 8 個の 4 面体に分割される。

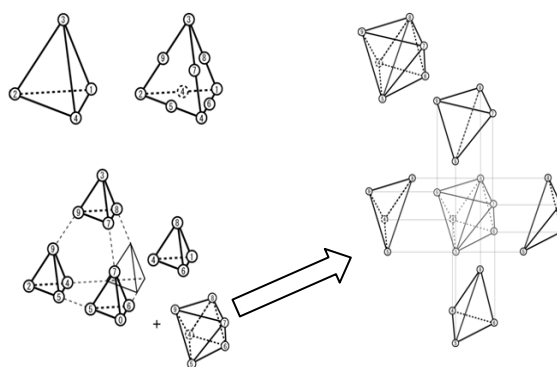


図 1

1.2. 6 面体 1 次要素の細分

6 面体については以下のようにして細分される。

- ① それぞれの辺の中点、面の中心、要素全体の中心に節点を追加する
- ② 元の 6 面体の頂点を含むような 6 面体 8 個に分割する。

*アドバンスソフト株式会社 技術第 5 部

5th Technical Division, AdvanceSoft Corporation

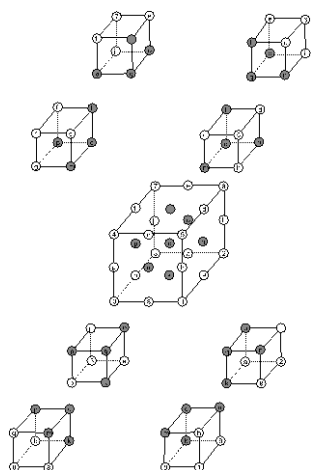


図 2

1.3. 3角柱 1次要素の細分

3角柱については以下のように細分される

- ① それぞれの辺の中点と4角形の面の中心に節点を追加する
- ② 4角形の面の中心の3つの節点を通るような面で元の3角柱を2つに分割する
- ③ 分割した3角柱を、3角形の面の分割を上面、底面に適用して4つの3角柱に分割する

結果として8つの3角柱が得られる。

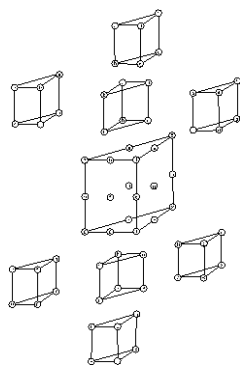


図 3

1.4. 4角錐 1次要素の細分

4角錐の細分は以下のように行う。

- ① それぞれの辺の中点と、底面の4角形の中心に節点を追加する。
- ② 元の4角錐の頂点を含むような5つの4角錐、底面の4角形の中心と側面の3角形の3つの中

間節点による4つの4面体、底面の4角形の中心を天頂とするような4角錐に分割する。

結果として6つの4角錐と4つの4面体の合計10個の要素が得られる。

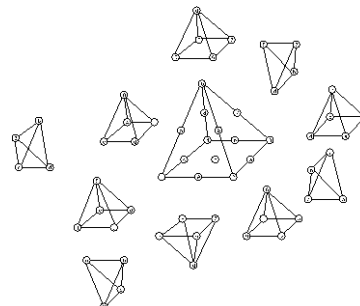


図 4

1.5. 境界条件の更新

REVOCAP_Refiner は要素の細分の前に境界条件を与えておくと、細分するときに同時に更新することができる。更新できる境界条件の種類は次の通りである。

- 節点グループ
- 要素グループ
- 面グループ（要素番号と要素内面番号の組）

FrontISTR の節点拘束、面荷重（一定値）の境界条件については上記の境界条件で対応可能である。境界条件の更新は要素の細分と同時に行われる。節点グループの更新規則は、周辺の節点がすべて節点グループに含まれる時に、中間節点もその節点グループに含まれるものとする（すなわち論理積を取る）。

1.6. 通信テーブルの更新

ソルバーが並列で解析する場合に通信テーブルの更新については、REVOCAP_Refiner が直接関与することはなく、REVOCAP_Refiner を組み込んだソルバーが処理を記述するものとし、REVOCAP_Refiner はそれに必要な関数群を提供するにとどめる。ここで想定されている関数群は、中間節点から元の親節点を問い合わせる関数などである。

1.7. REVOCAP_Refiner における形状補正機能

REVOCAP_Refiner の形状補正機能では、細分前の疎なメッシュにおいて、CAD の形状データと疎なメッシュの表面との間に対応関係を生成しておく（前処理）、解析モデルの細分の際の中間節点の生成の時に、幾何的な中点ではなく、CAD の形状データが与える曲面のパラメータを使って中点を求めるものである。CAD の曲面データとしては NURBS に対応している。

以下は形状補正機能を利用して解析モデルを細分した例である。

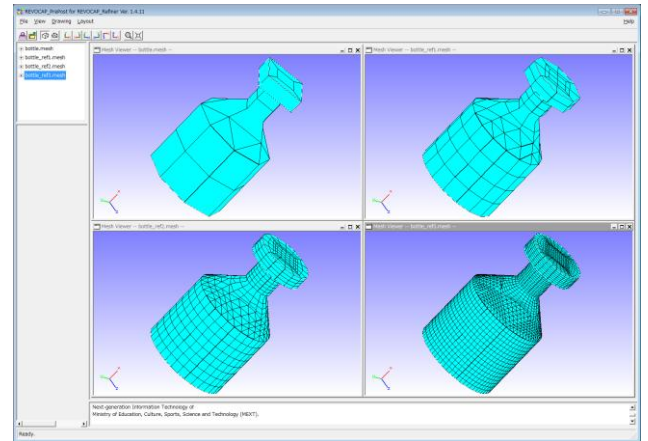


図 7 ボトル（要素タイプ混合の場合）

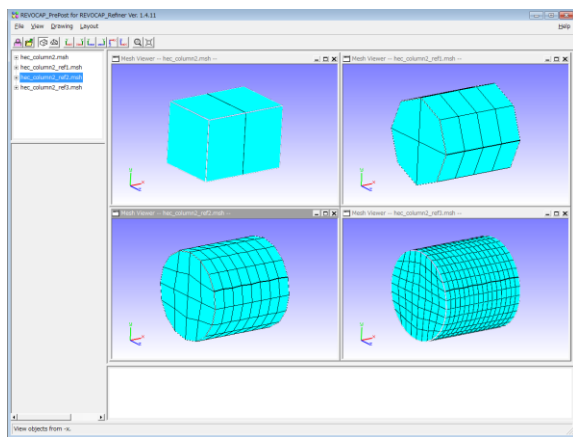


図 5 円柱

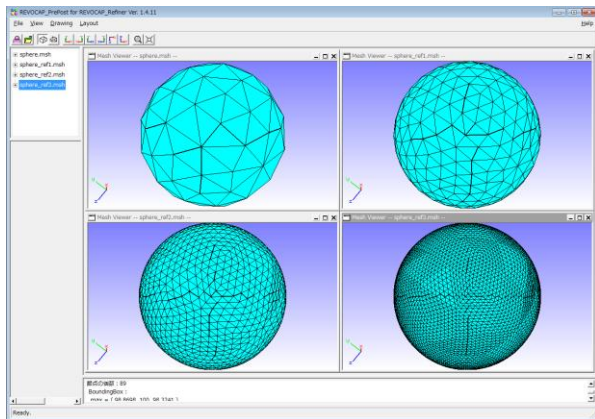


図 6 球（オートメッシュで生成したものを疎メッシュとした場合）

2. 異なる要素タイプへの分割（その 1: 4 面体への分割）

6 面体、3 角柱、4 角錐のそれぞれ 1 つの要素について 4 面体分割する方法は自明であるが、メッシュ全体として 4 面体分割する場合は隣接する要素と境界面のトポロジーが一致するように分割する必要がある。3 角柱については表面の分割の仕方によっては 4 面体分割できない有名な反例

（Schonhardt の多面体）があるので、それに抵触しないように全体のトポロジーを決定する必要があることに注意する。

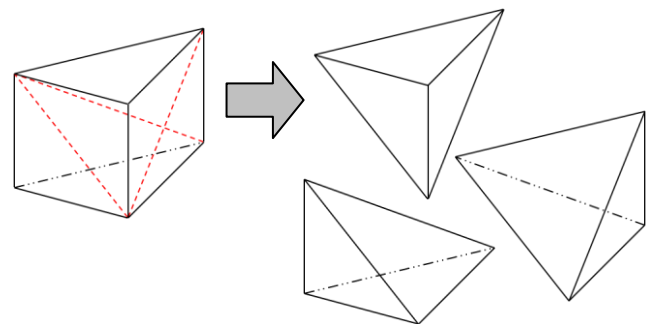


図 8 4 面体分割できる場合

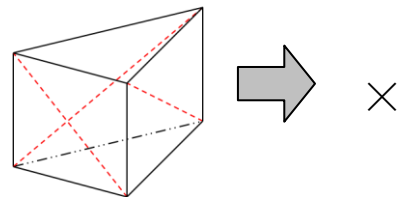


図 9 4 面体分割できない場合
（Schonhardt の多面体）

2.1.1. 補題(1)

6 面体、3 角柱、4 角錐について

- それぞれの頂点に整列順序関係が与えられている
- 表面の 4 角形にその頂点の整列順序関係での最小元を通る対角線を加えて 2 つの 3 角形に分割する

この条件でメッシュ全体についてそれぞれの要素の表面 4 角形を 3 角形に分割すれば、全体で境界面のトポロジーは適合している。

3次元空間上のメッシュの全節点について整列順序関係を与えることができるのは自明である。実装では順序関係は数値誤差を考慮して幾何的な条件で与えるのではなく、位相的な条件で与えるべきである。

2.1.2. 補題(2)

補題(1)のようにそれぞれの要素の表面 4 角形を 3 角形に分割した場合に、表面に適合するような 4 面体への分割が存在する。

(すなわち、3 角柱の場合に Schonhardt の反例に抵触しない)

4 角錐の場合は自明である。3 角柱の場合は 6 つの頂点のうち整列順序関係について最小のものを考えると、2 本の対角線がその点を通るため、Schonhardt の反例にはならない。

6 面体の場合は、8 つの頂点のうち整列順序関係について最小のものを考えると、3 本の対角線がその点を通る。これらの対角線が与えられた面の対面に同じ方向の対角線が与えられる場合は、6 面体は 2 つの 3 角柱に分割され、3 角柱は上記の方法で分割される。

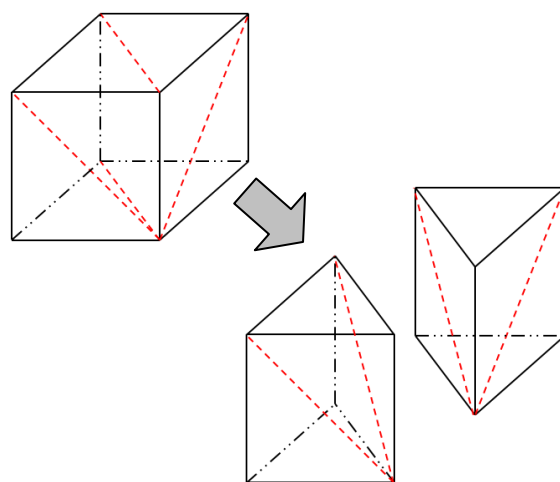


図 10

対角線が与えられた 3 つの面の対面にすべて交差する方向の対角線が与えられる場合は、5 つの 4 面体へに分割される。

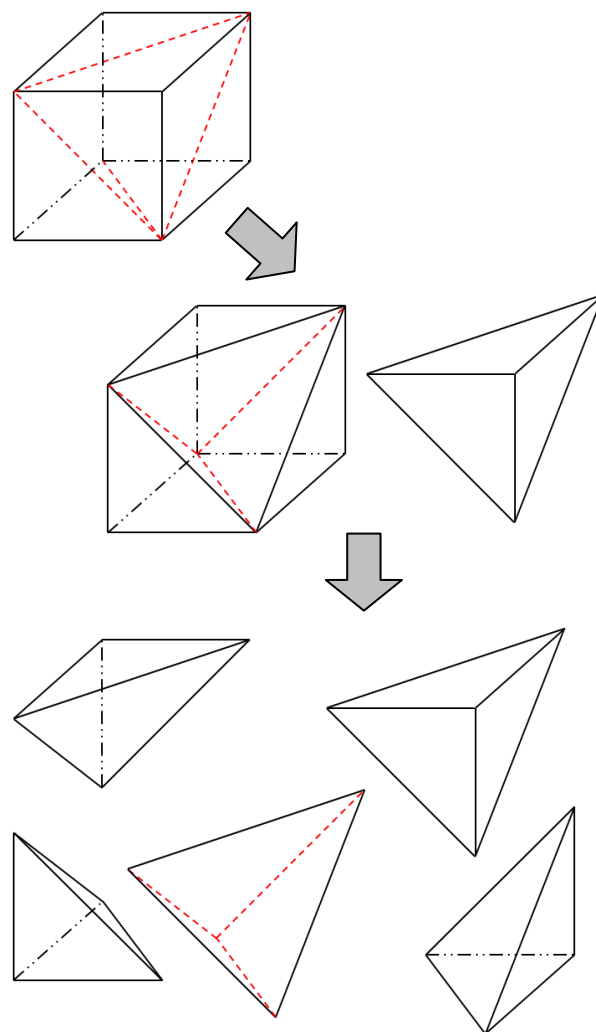


図 11

2.1.3. 定理

6 面体、3 角柱、4 角錐、4 面体からなるメッシュについて、節点を追加することなく全ての要素を境界面が適合した 4 面体に分割することができる。

3. 異なる要素タイプへの分割 (その 2: 6 面体への分割)

4 面体、3 角柱、6 面体、4 角錐を 6 面体に分割する方法について概説する。

例えば 1 つの 4 面体の場合、それぞれの表面の 3 角形を、辺上に新たな 3 つの頂点と 3 角形の中心に 1 つの頂点を追加することで 3 つの 4 角形に分割することができる。4 面体の中央に頂点を 1 つ追加することで、この表面の 4 角形に適合するような 4 つの 6 面体に分割することができる。

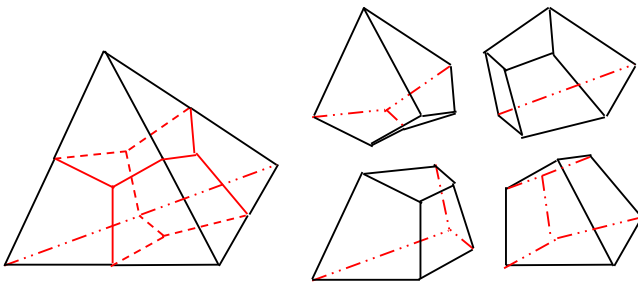


図 12

3 角柱の場合は 3 角形の面については 4 面体と同様に 3 つの 4 角形に分割し、4 角形の面についてはそれぞれの辺の中点と中心に頂点を追加して 4 つの 4 角形に分割し、これらの面に適合するように 6 つの 6 面体に分割することができる。

6 面体の場合は上で述べた細分と同じ方法で 8 つの 6 面体に分割することができる。

4 面体、3 角柱、6 面体の場合はいずれも表面の 3 角形、4 角形を、各辺の中点および重心に点を与えて 3 つまたは 4 つの 4 角形に分割している。

4 角錐の場合に、表面を同様に 4 角形に分割した場合に内部を 6 面体で埋めることができるかどうか、という問題は Schneiders' Open Problem として知られているが、Yamakawa and Shimada [6] において HEXHOOP 法と呼ばれる幾つかのテンプ

レートを組み合わせる方法で解かれた。以下ではこの方法を別の見方で見直しながら紹介する。

3.1.1. 定義

4 角形からなる境界付きの境界に基点を持つ向きづけられた表面メッシュ Q_1 と Q_2 があるとき、

- Q_1 と Q_2 の境界上の節点数が一致する
- 基点を一致させて Q_1 と Q_2 の境界を順番につないでできる 4 面体メッシュの内部を 6 面体で埋めることができる

場合に組み合わせ的 6 面体コボルダント

(Combinatorial Hexahedral Cobordant=CHC) と呼ぶことにする。

この定義は、このノート独自の定義である。

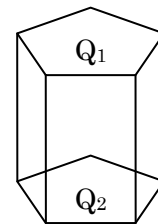


図 13

3.1.2. 定理

CHC 関係は同値関係である

3.1.3. 証明

反射律と対称律は自明である。推移律も次のようなメッシュを考えれば Q_1 と Q_2 が CHC で Q_2 と Q_3 が CHC のときに Q_1 と Q_3 が CHC なのは明らかである。

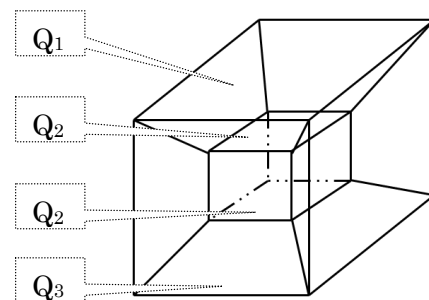


図 14

今後は CHC 関係を $\sim$ で表すことにする。

3.1.4. 例(1) : 上の証明で出てくる図形を Q_2 のところで切断する

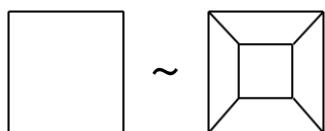


図 15

3.1.5. 例(2) : 4 面体を 6 面体に分割したものの表面

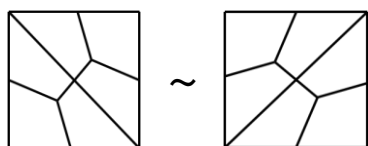


図 16

3.1.6. 例(3) : 3 角柱を 6 面体に分割したものの表面

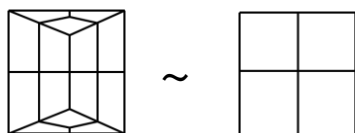


図 17

3.1.7. 例(4) : HEXHOOP-Template

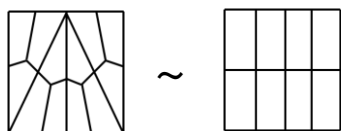


図 18

3.1.8. 定理

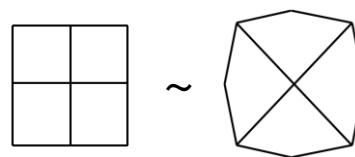


図 19

3.1.9. 系

4 角錐は以下のように表面を 4 角形に分割した場合、内部を 6 面体で埋めることができる

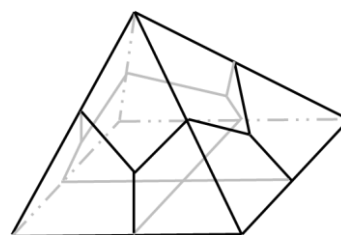


図 20

3.1.10. 系の証明

底面は定理の左辺と等しく、上部の頂点を含む 4 角形は定理の右辺に等しい。両者の境界の頂点を順に結んだものが 4 角錐の表面を 4 角形に分割したメッシュを与える。従って上記の定理は 4 角錐の内部を 6 面体で埋めることができることを表している。

定理の証明は構成論的に[6]で与えられている。ここではそこで与えられているものを CHC 関係で表したものを示す。

3.1.11. 定理の証明の概略

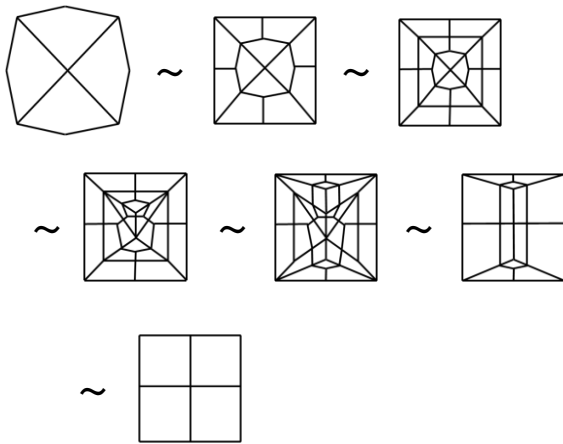


図 21

参考文献

- [1] Joe F. Thompson, Bharat K. Soni and Nigel P. Weatherill, Handbook of Grid Generation (CRC Press), 1998
- [2] Pascal Jean Frey and Paul-Louis George, Mesh Generation second edition (John Wiley & Sons, Inc), 2008
- [3] Scott A. Mitchell, A Characterization of the Quadrilateral Meshes of a Surface Which Admit a Compatible Hexahedral Mesh of the Enclosed Volume, Lecture Notes in Computer Science, 1996, Volume 1046/1996, 465-476
- [4] T.J.Tautges and S. Mitchell, Progress report on the whisker weaving all hexahedral meshing algorithm, Numerical grid generation in computational field simulations; Proceedings of the 5th International Conference, Mississippi State Univ, Mississippi State; UNITED STATES; 1-5 Apr. 1996. pp. 659-670. 1996
- [5] T.J.Tautges, T.Blackner and Scott A. Mitchell, The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All HExahedral Finite Element Meshes, Int. J. Numer. Methods Eng., 39:3327-3349,

(1996)

- [6] Yamakawa, S. and K. Shimada, "HEXHOOP: Modular Templates for Converting a Hex-Dominant Mesh to an ALL-Hex Mesh," The 10th International Meshing Roundtable, 2001
- [7] S. A. Mitchell, "The All-Hex Geode-Template for Conforming a Diced Tetrahedral Mesh to any Diced Hexahedral Mesh," The 7th International Meshing Roundtable, 1998.