

文部科学省次世代 I T 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」

CISS フリーソフトウェア

マルチ力学シミュレータ REVOCAP

モデル細分化ツール

REVOCAP_Refiner Ver. 0.4

ユーザマニュアル

本ソフトウェア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」プロジェクトによる成果物です。本ソフトウェアを無償でご使用になる場合「CISS フリーソフトウェア使用許諾条件」をご了承頂くことが前提となります。営利目的の場合には別途契約の締結が必要です。これらの契約で明示されていない事項に関して、或いは、これらの契約が存在しない状況においては、本ソフトウェアは著作権法など、関係法令により、保護されています。

お問い合わせ先

(契約窓口)

(財)生産技術研究奨励会

〒153-8505 東京都目黒区駒場4-6-1

(ソフトウェア管理元)

東京大学生産技術研究所 革新的シミュレーション研究センター

〒153-8505 東京都目黒区駒場4-6-1

Fax : 03-5452-6662

E-mail : software@ciss.iis.u-tokyo.ac.jp

目次

1	概要	2
2	動作環境	2
3	開発言語・利用しているライブラリ	2
4	インストール	3
5	要素ライブラリ	3
5.1	1次線分要素(SEGMENT)	3
5.2	2次線分要素 (SEGMENT2)	4
5.3	1次三角形要素 (TRIANGLE)	4
5.4	2次三角形要素 (TRIANGLE2)	5
5.5	1次四角形要素 (QUAD)	5
5.6	2次四角形要素 (QUAD2)	6
5.7	1次四面体要素 (TETRAHEDRON)	7
5.8	2次四面体要素 (TETRAHEDRON2)	8
5.9	1次六面体要素 (HEXAHEDRON)	8
5.10	2次六面体要素 (HEXAHEDRON2)	9
5.11	1次三角柱要素 (WEDGE)	11
5.12	2次三角柱要素 (WEDGE2)	12
5.13	1次四角錐要素 (PYRAMID)	13
5.14	2次四角錐要素 (PYRAMID2)	14
6	解析モデルの細分	15
6.1	四面体1次要素の細分	15
6.2	六面体1次要素の細分	18
6.3	三角柱1次要素の細分	20
6.4	四角錐1次要素の細分	22
6.5	三角形1次要素の細分	22
6.6	線分1次要素の細分	22
6.7	境界条件	23
6.8	節点グループの更新規則	23

1 概要

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、カップラに組み込まれて使われる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である。

メッシュ生成、境界条件設定等のプリ処理済みの解析モデルについて、ソルバまたはカップラがファイルから読み込んでメモリ上に保持された状態で、REVOCAP_Refiner が提供する機能により、ソルバまたはカップラがメモリ上に持つ要素を細分する。したがって要素の入出力ともファイルを介さない。

単純に細分するだけでは、CAD モデルに対する形状の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。ただし β 版ではこの機能は利用できない。

2 動作環境

REVOCAP_Refiner の動作環境は、組み込まれて利用されるソルバおよびカップラの動作環境に準ずる。超並列計算機および次世代スパコンの上での動作を想定している。各種スパコン、PC クラスタ、次世代スパコン上 Unix 互換の OS のもとで動作する。動作が確認されている環境は以下の通り。

- SGI Altix
- Linux (CentOS)

3 開発言語・利用しているライブラリ

REVOCAP_Refiner は REVOCAP_PrePost とメッシュ処理部を共通化している。その部分の開発は C++ 言語でなされている。ソルバーとのインターフェイスを取る部分は C 言語で開発されている。言語標準の以外の外部ライブラリについては特に利用していない。動作が確認されているコンパイラは以下の通り。

- g++
- Intel C
- PGI C
- gfortran

- Intel Fortran
- PGI Fortran

4 インストール

提供されている `REVOCAP_Refiner_XXX.tgz` (XXX はバージョンの情報)を展開する。

`OPTIONS` ファイルを必要に応じて編集してください。

`make Refiner` を実行してください。`Makefile` は `PrePost` 用の拡張ライブラリ作成のためのものと共通のため、他にもいくつか `make` ターゲットが定義されていますが、`REVOCAP_Refiner` に関係があるものは以下の通りです。

- `Refiner` : `REVOCAP_Refiner` のライブラリをビルドします
- `RefinerSampleC` : `REVOCAP_Refiner` の C 言語サンプルをビルドします
- `RefinerSampleF` : `REVOCAP_Refiner` の Fortran 言語サンプルをビルドします
- `RefinerDoc` : `REVOCAP_Refiner` のドキュメントを生成します。`Doxygen` がインストールされている必要があります。
- `RefinerClean` : `REVOCAP_Refiner` に関連する生成オブジェクトを削除します

5 要素ライブラリ

以下は `REVOCAP_PrePost` のメッシュ処理部で内部データとして保持することのできる要素の一覧である。この節点配列の順番で要素の登録、要素データの取得が行われる。以下の記述で節点配列や接続行列の添え字は 0 から始まる。

接続行列は i 番目の節点と j 番目の節点の接続情報を表している。ただし利用者が直接この行列を利用することはない。1 次要素とみなしてつながっているときは ± 1 とし、2 次要素とみなしてつながっているときは ± 2 とする。

符号は向きを表し、線要素、平面要素のみ有効である。

辺、面情報は内部で保持している辺や面の順番を記述してある。各行は立体要素の場合はその面、平面要素の場合はその辺の節点の接続情報が記述されている。(4 面体要素なら、それぞれの行が三角形要素の節点 ID の配列) 面は外から見て反時計回りが正となるように向き付けられている。

5.1 1 次線分要素(SEGMENT)

接続行列

0	1
-1	0

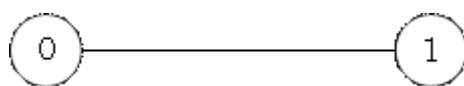


図 1 1次線分要素 (SEGMENT)

5.2 2次線分要素 (SEGMENT2)

接続行列

0	1	2
-1	0	-2
-2	2	0

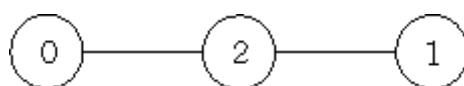


図 2 2次線分要素 (SEGMENT2)

5.3 1次三角形要素 (TRIANGLE)

接続行列

0	1	-1
-1	0	1
1	-1	0

辺

1	2
2	0
0	1

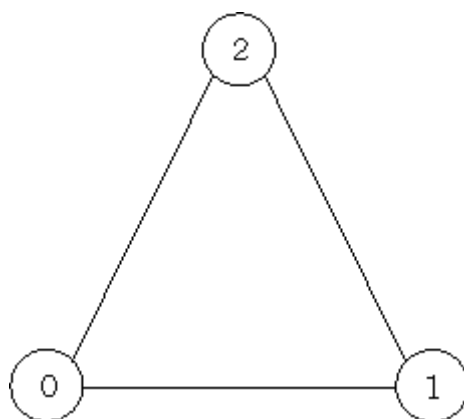


图 3 1 次三角形要素 (TRIANGLE)

5.4 2 次三角形要素 (TRIANGLE2)

接續行列

0	1	-1	0	-2	2
-1	0	1	2	0	-2
1	-1	0	-2	2	0
0	-2	2	0	0	0
2	0	-2	0	0	0
-2	2	0	0	0	0

辺

1	2	3
2	0	4
0	1	5

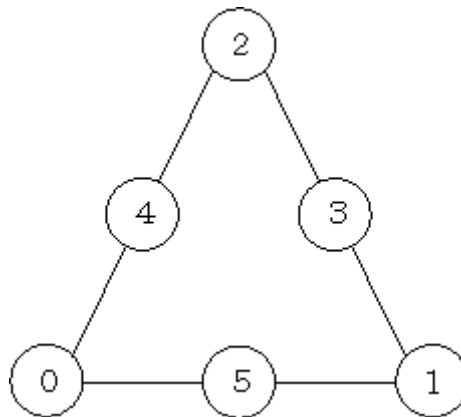


图 4 2 次三角形要素 (TRIANGLE2)

5.5 1 次四角形要素 (QUAD)

接續行列

0	1	0	-1
-1	0	1	0
0	-1	0	1
1	0	-1	0

辺

0	1
---	---

1	2
2	3
3	0

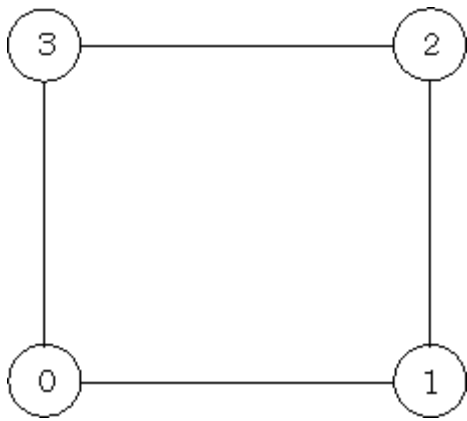


圖 5 1 次四角形要素 (QUAD)

5.6 2 次四角形要素 (QUAD2)

接續行列

0	1	0	-1	2	0	0	-2
-1	0	1	0	-2	2	0	0
0	-1	0	1	0	-2	2	0
1	0	-1	0	0	0	-2	2
-2	2	0	0	0	0	0	0
0	-2	2	0	0	0	0	0
0	0	-2	2	0	0	0	0
2	0	0	-2	0	0	0	0

邊

0	1	4
1	2	5
2	3	6
3	0	7

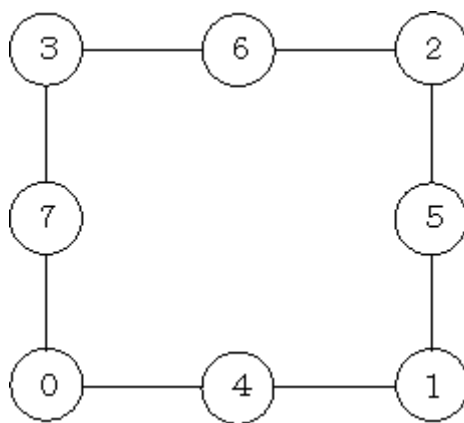


图 6 2 次四角形要素 (QUAD2)

5.7 1 次四面体要素 (TETRAHEDRON)

接續行列

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

面

1	2	3
0	3	2
0	1	3
0	2	1

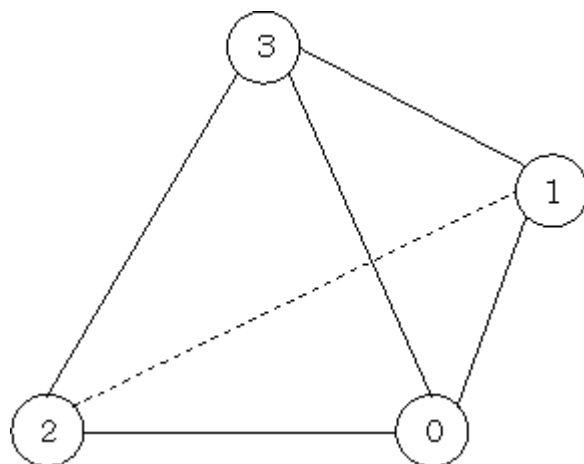


図 7 1 次四面体要素 (TETRAHEDRON)

5.8 2 次四面体要素 (TETRAHEDRON2)

接続行列

0	1	1	1	0	2	2	2	0	0
1	0	1	1	2	0	2	0	2	0
1	1	0	1	2	2	0	0	0	2
1	1	1	0	0	0	0	2	2	2
0	2	2	0	0	0	0	0	0	0
2	0	2	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0
2	0	0	2	0	0	0	0	0	0
0	2	0	2	0	0	0	0	0	0
0	0	2	2	0	0	0	0	0	0

面情報

1	2	3	9	8	4
0	3	2	9	5	7
0	1	3	8	7	6
0	2	1	4	6	5

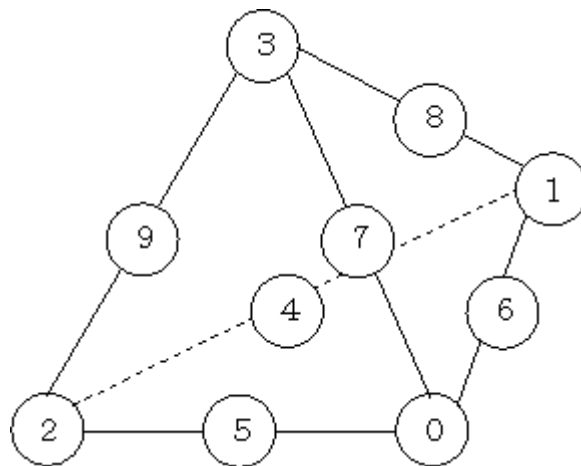


図 8 2 次四面体要素 (TETRAHEDRON2)

5.9 1 次六面体要素 (HEXAHEDRON)

接続行列

0	1	0	1	1	0	0	0
1	0	1	0	0	1	0	0
0	1	0	1	0	0	1	0
1	0	1	0	0	0	0	1
1	0	0	0	0	1	0	1
0	1	0	0	1	0	1	0
0	0	1	0	0	1	0	1
0	0	0	1	1	0	1	0

面

3	2	1	0
4	5	6	7
1	5	4	0
1	2	6	5
3	7	6	2
4	7	3	0

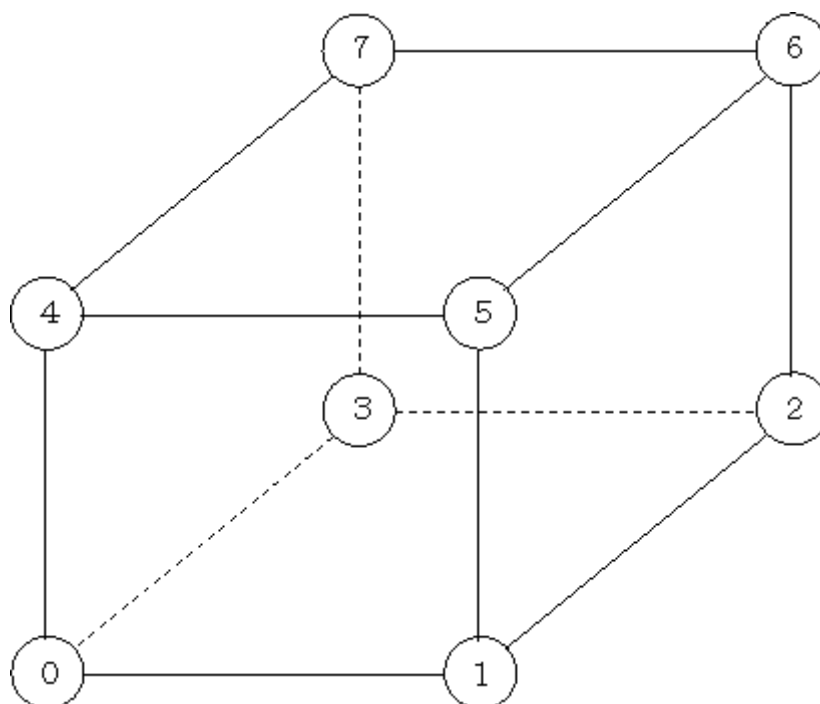


図 9 1 次六面体要素 (HEXAHEDRON)

5.102 次六面体要素 (HEXAHEDRON2)

接続行列

0	1	0	1	1	0	0	0	2	0	0	2	0	0	0	0	2	0	0	0
1	0	1	0	0	1	0	0	2	2	0	0	0	0	0	0	0	2	0	0
0	1	0	1	0	0	1	0	0	2	2	0	0	0	0	0	0	0	2	0
1	0	1	0	0	0	0	1	0	0	2	2	0	0	0	0	0	0	0	2
1	0	0	0	0	1	0	1	0	0	0	0	2	0	0	2	2	0	0	0
0	1	0	0	1	0	1	0	0	0	0	0	2	2	0	0	0	2	0	0
0	0	1	0	0	1	0	1	0	0	0	0	0	2	2	0	0	0	2	0
0	0	0	1	1	0	1	0	0	0	0	0	0	0	2	2	0	0	0	2
2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0

面

3	2	1	0	10	9	8	11
4	5	6	7	12	13	14	15
1	5	4	0	17	12	16	8
1	2	6	5	9	18	13	17
3	7	6	2	19	14	18	10
4	7	3	0	15	19	11	16

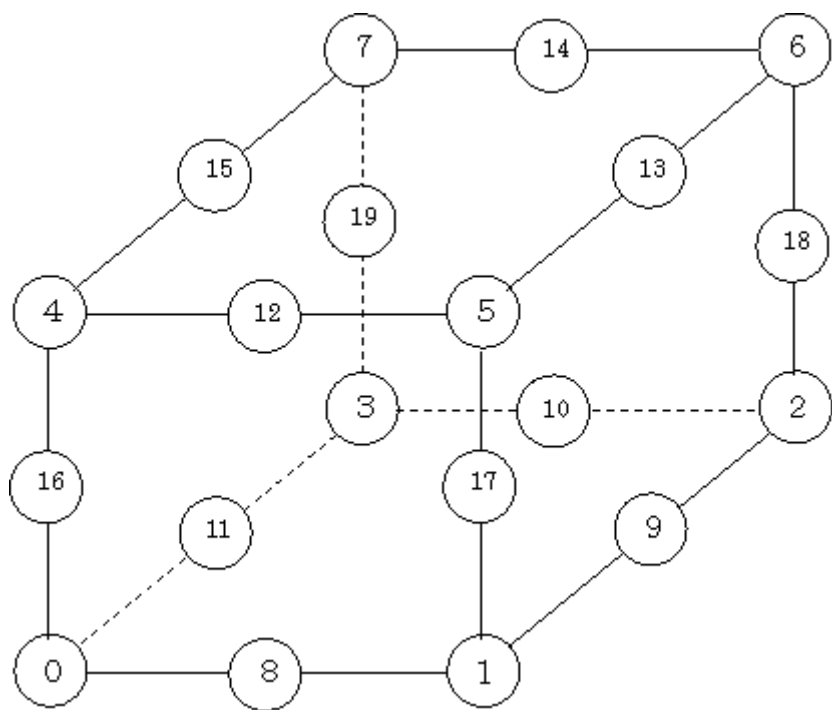


図 10 2次六面体要素 (HEXAHDRON2)

5.111 次三角柱要素 (WEDGE)

接続行列

0	1	1	1	0	0
1	0	1	0	1	0
1	1	0	0	0	1
1	0	0	0	1	1
0	1	0	1	0	1
0	0	1	1	1	0

面情報

0	2	1	
3	4	5	
0	1	4	3
1	2	5	4
2	0	3	5

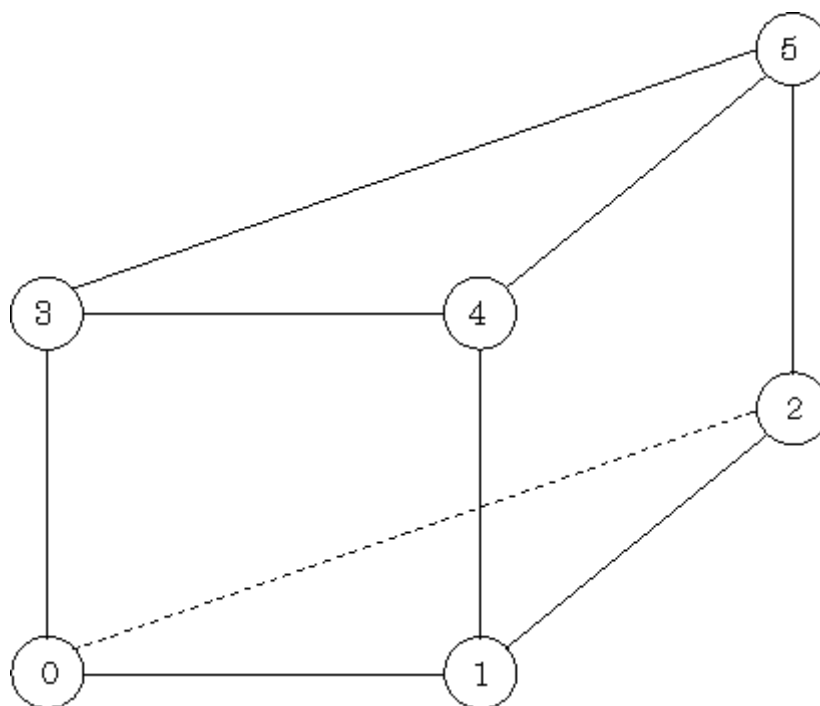


图 11 1 次三角柱要素 (WEDGE)

5.122 次三角柱要素 (WEDGE2)

接續行列

0	1	1	1	0	0	0	2	2	0	0	0	2	0	0
1	0	1	0	1	0	2	0	2	0	0	0	0	2	0
1	1	0	0	0	1	2	2	0	0	0	0	0	0	2
1	0	0	0	1	1	0	0	0	0	2	2	2	0	0
0	1	0	1	0	1	0	0	0	2	0	2	0	2	0
0	0	1	1	1	0	0	0	0	2	2	0	0	0	2
0	2	2	0	0	0	0	0	0	0	0	0	0	0	0
2	0	2	0	0	0	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	2	0	0	0	0	0	0	0	0	0
0	0	0	2	0	2	0	0	0	0	0	0	0	0	0
0	0	0	2	2	0	0	0	0	0	0	0	0	0	0
2	0	0	2	0	0	0	0	0	0	0	0	0	0	0
0	2	0	0	2	0	0	0	0	0	0	0	0	0	0
0	0	2	0	0	2	0	0	0	0	0	0	0	0	0

面情報

0	2	1	6	8	7		
3	4	5	9	10	11		
0	1	4	3	8	13	11	12
1	2	5	4	6	14	9	13
2	0	3	5	7	12	10	14

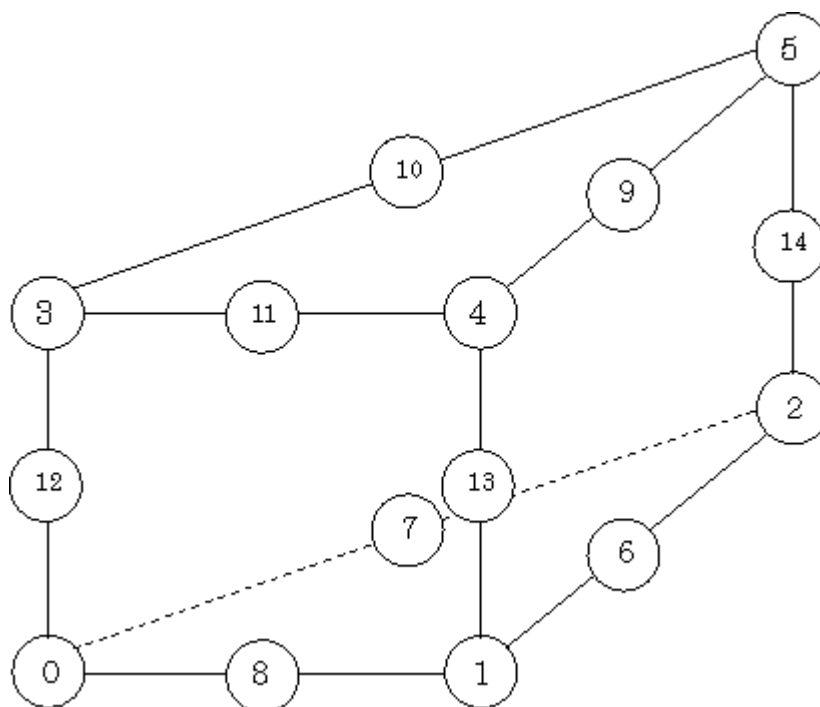


図 12 2次三角柱要素 (WEDGE2)

5.131 次四角錐要素 (PYRAMID)

接続行列

0	1	1	1	1
1	0	1	0	1
1	1	0	1	0
1	0	1	0	1
1	1	0	1	0

面

0	1	2	
0	2	3	
0	3	4	

0	4	1	
4	3	2	1

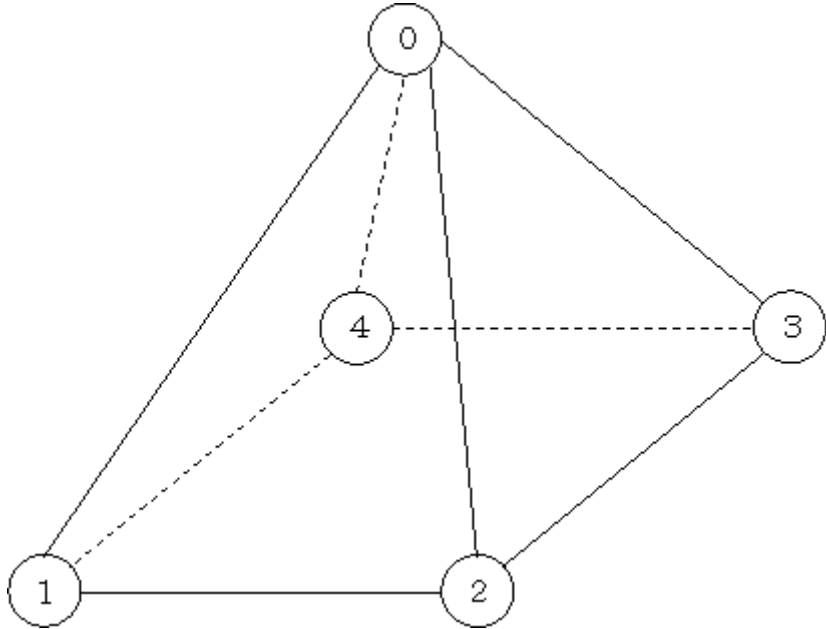


圖 13 1 次四角錐要素 (PYRAMID)

5.142 次四角錐要素 (PYRAMID2)

接續行列

0	1	1	1	1	2	2	2	2	0	0	0	0
1	0	1	0	1	2	0	0	0	2	0	0	2
1	1	0	1	0	0	2	0	0	2	2	0	0
1	0	1	0	1	0	0	2	0	0	2	2	0
1	1	0	1	0	0	0	0	2	0	0	2	2
2	2	0	0	0	0	0	0	0	0	0	0	0
2	0	2	0	0	0	0	0	0	0	0	0	0
2	0	0	2	0	0	0	0	0	0	0	0	0
2	0	0	0	2	0	0	0	0	0	0	0	0
0	2	2	0	0	0	0	0	0	0	0	0	0
0	0	2	2	0	0	0	0	0	0	0	0	0
0	0	0	2	2	0	0	0	0	0	0	0	0
0	2	0	0	2	0	0	0	0	0	0	0	0

面

0	1	2	9	6	5		
0	2	3	10	7	6		
0	3	4	11	8	7		
0	4	1	12	5	8		
4	3	2	1	11	10	9	12

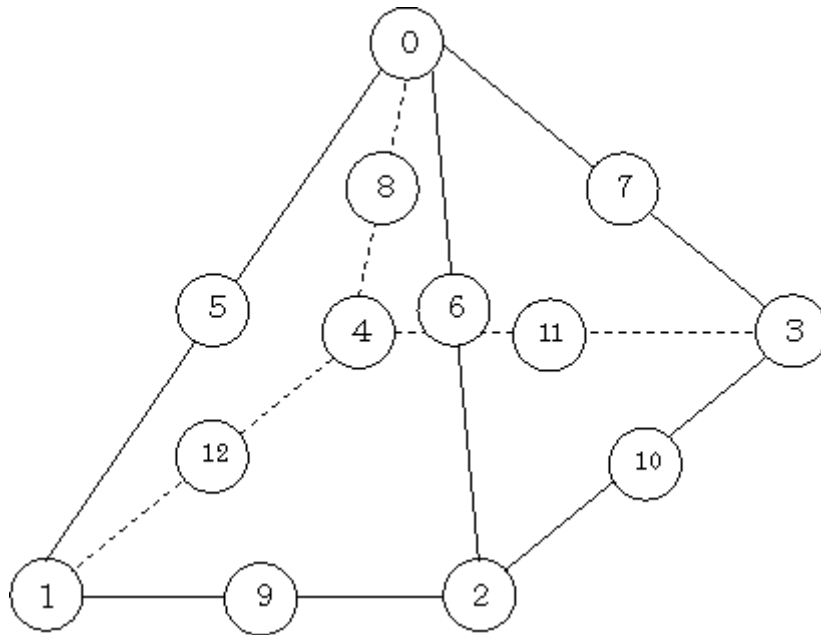


図 14 2次四角錐要素 (PYRAMID2)

6 解析モデルの細分

6.1 四面体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) 元の四面体の頂点を含むような4つの四面体と中央の八面体に分割する
- (3) 中央の八面体について最も長さの短い対角線を追加して、4つの四面体に分割する
結果として8個の四面体に分割される。

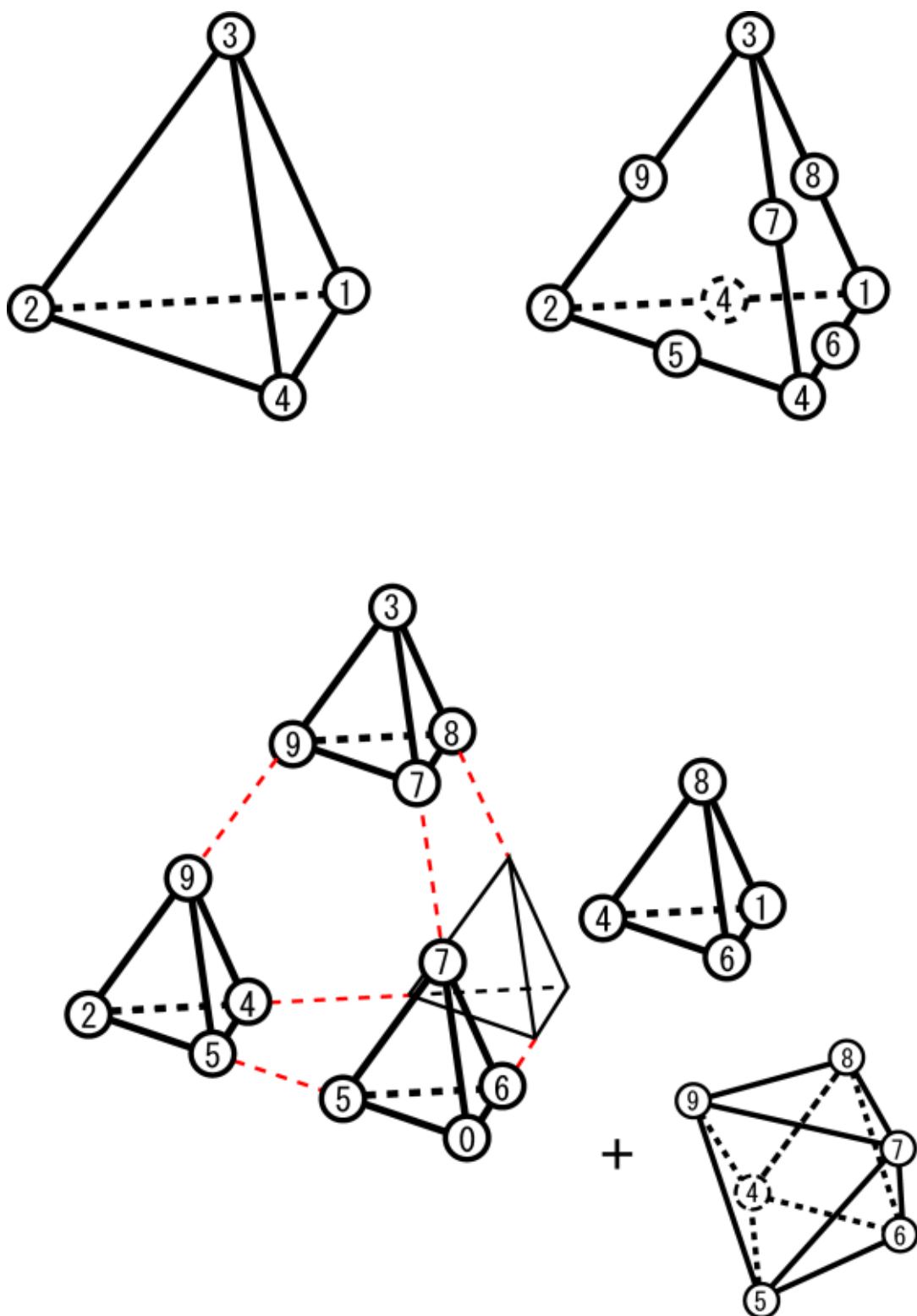


図 6.1.1 4 面体細分 (1)

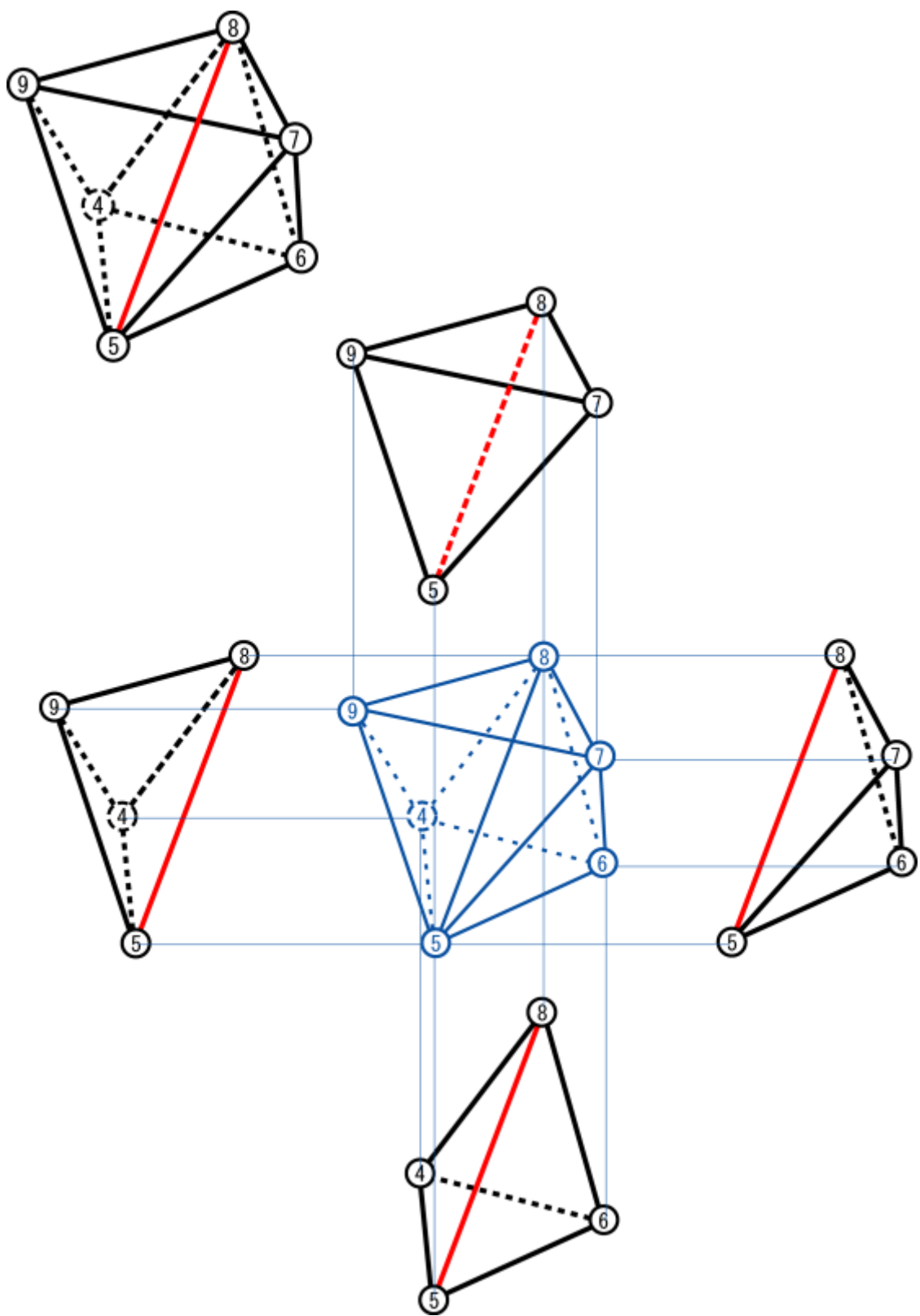
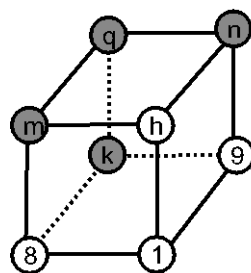
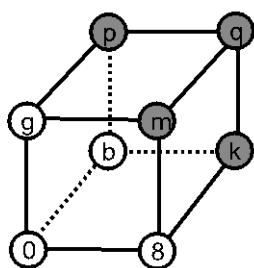
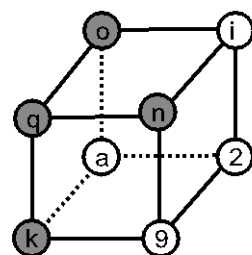
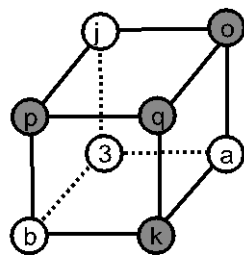
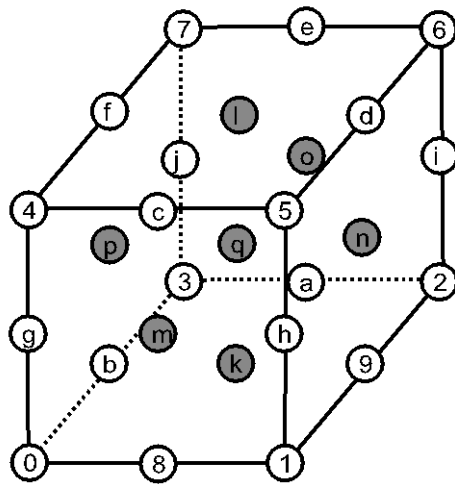
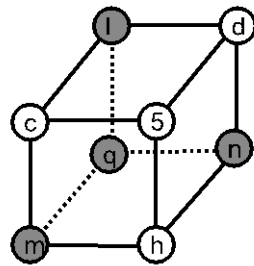
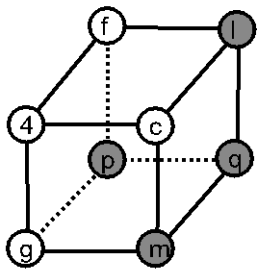
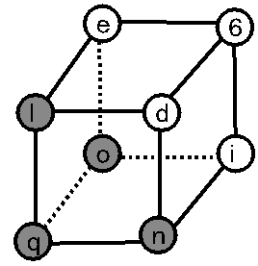
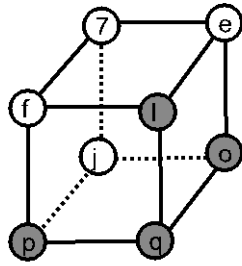


図 6.1.2 4 面体細分 (2)

6.2 六面体 1 次要素の細分

六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
元の六面体の頂点を含むような六面体 8 個に分割する。

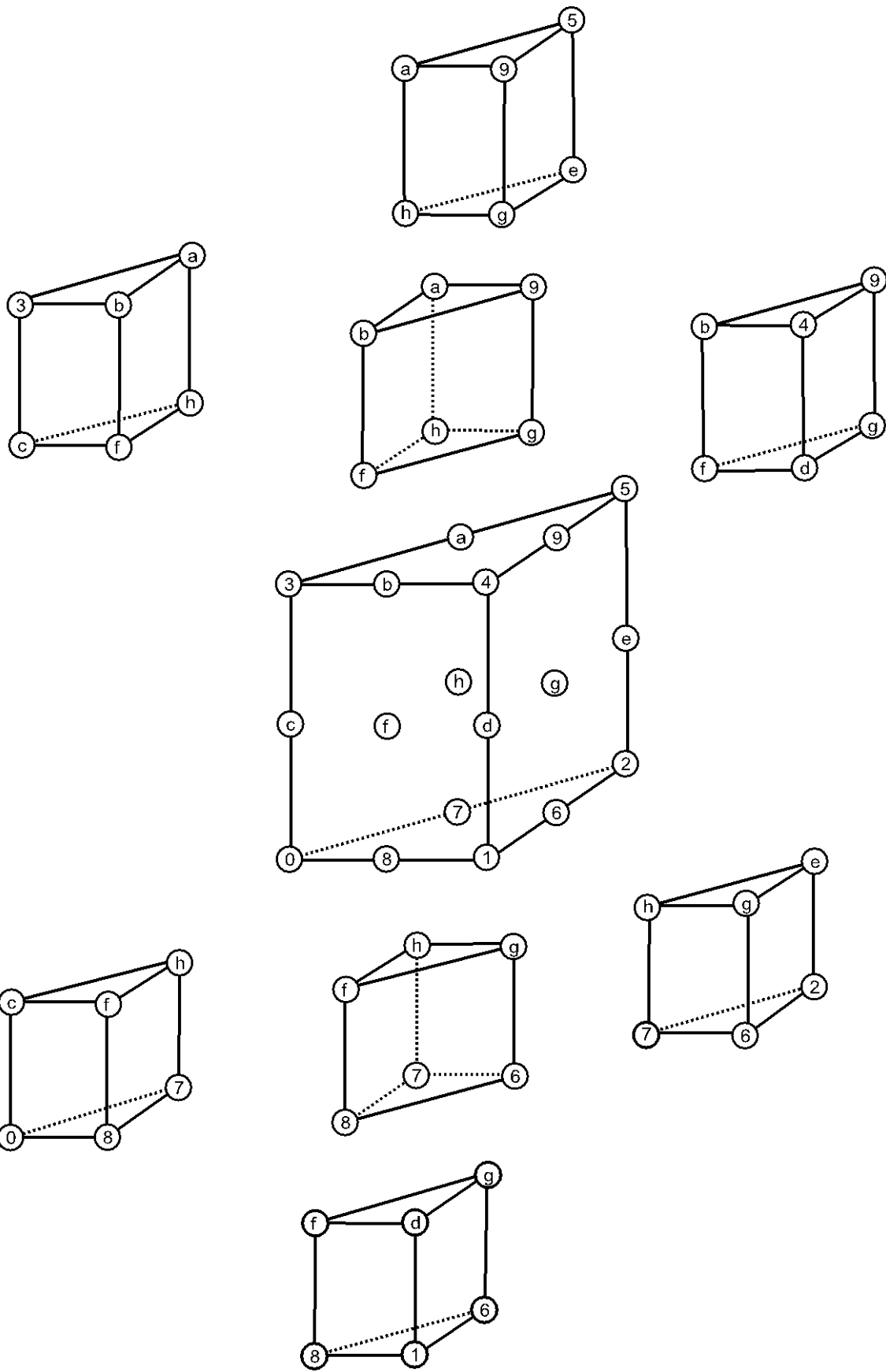


6.3 三角柱 1 次要素の細分

三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四角形の面の中心に節点を追加する
- (2) 四角形の面の中心の 3 つの節点を通るような面で元の三角柱を 2 つに分割する
- (3) 分割した三角柱を、三角形の面の分割を上面、底面に適用して 4 つの三角柱に分割する

結果として 8 つの三角柱が得られる。

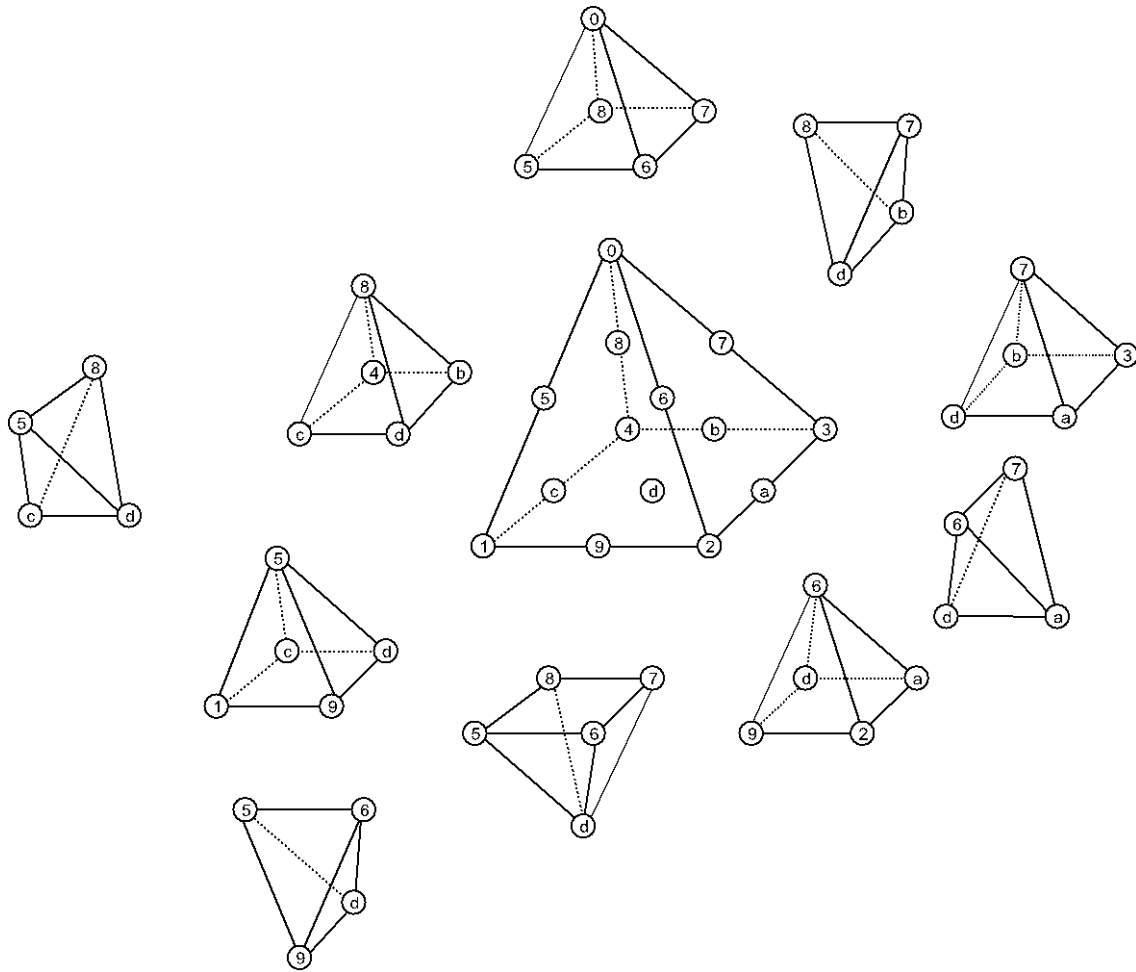


6.4 四角錐 1 次要素の細分

四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、底面の四角形の中心に節点を追加する。
- (2) 元の四角錐の頂点を含むような 5 つの四角錐、底面の四角形の中心と側面の三角形の 3 つの中間節点による 4 つの四面体、底面の四角形の中心を天頂とするような四角錐に分割する。

結果として 6 つの四角錐と 4 つの四面体の合計 10 個の要素が得られる。



6.5 三角形 1 次要素の細分

それぞれの辺の中点に新たな節点を追加し、新たに 4 個の要素を生成する。

6.6 線分 1 次要素の細分

それぞれの辺の中点に新たな節点を追加し、新たに 2 個の要素を生成する。

6.7 境界条件

REVOCAP_Refiner は要素の細分の前に境界条件を登録しておく、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）

分布については対応していない。FrontSTR の節点拘束、面荷重（一定値）、FrontFlow/blue 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。

境界条件の更新のための関数があるのではなく、細分時の更新は要素の細分と同時に行われる。従って、境界条件の細分は以下の手順で行う。

- (1) 境界条件を Refiner に登録する(rcapAppendNodeGroup、rcapAppendElementGroup、rcapAppendFaceGroup)
 - (2) 細分前の節点を Refiner に登録する(rcapSetNodeXX)
 - (3) 要素を細分する(rcapRefineElement、rcapRefineElementMulti)
 - (4) 境界条件をコミットする(rcapCommit)
 - (5) 更新された境界条件を Refiner から取得する(rcapGetNodeGroup、rcapGetElementGroup、rcapGetFaceGroup)
- (1)と(2)は順序が逆でもよい。

複数の境界条件を同時に更新するために境界条件には名前をつけて登録する。更新後に取得するにはその名前を Refiner に与える。

6.8 節点グループの更新規則

節点グループの要素細分時の更新規則は以下の通りとする。

- (1) 要素の辺の midpoint に中間節点が与えられた場合は、その辺の両端の節点がともに節点グループに含まれる場合に限り、midpoint を節点グループに追加する。
- (2) 要素の面の中心に節点が与えられた場合は、その面の頂点の節点がすべて節点グループに含まれる場合に限り、中心を節点グループに追加する。
- (3) 要素の中心に節点が与えられた場合は、その要素の頂点の節点がすべて節点グループに含まれる場合に限り、中心を節点グループに追加する。

すなわち周辺の節点についての論理積を取る。