

3 次元単体メッシュ生成の課題——計算幾何学の立場から——

東京大学工学系研究科 杉原厚吉 (SUGIHARA Kokichi)

1 はじめに

メッシュは偏微分方程式の数値解法のためにはなくてはならないものであるが、その完全自動生成技術の開発は、有限要素法などの数値解法自身より遅れている。大規模な問題でも、離散化された方程式自身は数時間から数日で解けるのに、離散化を行なうための前処理であるメッシュの生成が人手にたよらざるを得ないため、1ヶ月から数ヶ月かかることも少なくない。このようにメッシュの生成は、偏微分方程式の数値解法のボトルネックとなっている [5, 11, 12, 13]。

メッシュ構造は、大きく分けて単体系のもの (2次元では三角形、3次元では四面体) と四角形・六面体系のものがある。後者は解の精度の面からはすぐれているが、一般に作ることが難しい。一方、前者のメッシュは、複雑な境界で囲まれた領域に対しても比較的作りやすく、自動化が相対的には容易である。実際、2次元領域の三角形メッシュの生成については自動化技術がほぼ完成している [5]。しかし、3次元領域の四面体メッシュの生成に関しては、まだ問題が山積している。

そこで本稿では、3次元単体メッシュの構成法に焦点を当て、自動生成技術の現状と今後に残されている主な課題についてまとめる。

2 メッシュ生成問題とその可解性

任意の2次元多角形に対して、その頂点のみを頂点に用いて内部を三角形に分割することは常に可能である。一方、表面が三角形面だけで構成される任意の3次元多面体に対して、その頂点のみを頂点に用いて内部を四面体に分割する問題は、必ずしも解をもつとは限らない。

四面体分割のできない多面体の例を図1に示す。これは三角柱の側面の三つの四角形を、互いに端点を共有しないように対角線を入れて三角形に分割し、そのあと上面と底面を垂直な軸のまわりで逆向きに少し回転させたときできる立体である。側面に入れた3本の対角線がすべて立体の内部に食い込む形になっている。この立体の稜線でつながれていない2頂点を結ぶ線分は、すべて立体の外部を通過する。したがって、この立体を四面体に分割することはできない。分割するためには立体の内部に新しい頂点を設けて、それも使わなければならない。

境界が三角形面のみで構成された3次元領域 R (内部に穴があいていてもよい) と、 R のすべての頂点と R の内部に指定された有限個の点からなる集合 V が与えられたとき、 V に属する点のみを用いて R の内部を四面体に分割する問題を、四面体分割問題という。上で見たように、この問題は必ずしも解をもつとは限らない。また、解をもつ場合と解をもたない場合を区別する一般的方法も見つかっていない。この意味でも、3次元メッシュ生成問題は2次元の場合と比べて格段に難しい。

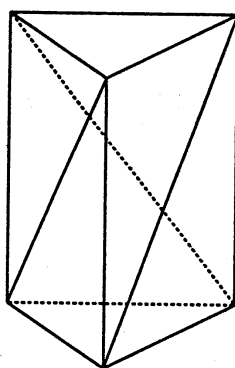


図 1. 四面体分割のできない多面体

3 形の良いメッシュの追求

偏微分方程式の数値解法に用いるためには、メッシュならどんなものでもよいというわけではない。解の精度を下げないためには“形の良い”メッシュが必要である。形の良いメッシュとは、一般的にはできるだけ“ふっくら”とした単体から構成されているメッシュである。場合によっては、2次元では鈍角三角形を含まない(直角三角形は含んでもよい)三角形メッシュや、鋭角三角形のみのメッシュが望まれたり [2, 4], 流体解析の場面では流れの方向に沿った細長い要素からなるメッシュが望まれることもある [21, 25].

良いメッシュを追求する問題は、使うべき頂点集合 V が指定された場面で考える場合と、使うべき頂点の数や位置を選ぶ自由度も利用して考える場合に分けられる。もちろん最終的には両者の技術を総合的に使うのであるが、まずは前者の問題を考えてみよう。

3.1 使うべき頂点が指定された場合

まず最も簡単な、2次元凸多面体の三角形分割を考える。すなわち、平面上に指定された有限個の点の集合 V に対して、 V に属す点を頂点として用いて V の凸包の内部を三角形に分割する問題を考える。同じ点集合を用いても、図 2 の (a), (b) に示すようにいろいろな分割の仕方がある。

ここで、できるだけ“ふっくら”とした三角形に分割したいという感覚的な表現を、もう少し数理的な言葉に翻訳しようとする、たとえば次のような諸基準が考えられよう。

(1) 最小角最大: 三角形の内角を小さい順に並べた列が辞書式順序で最大となる三角形分割を作る。

(2) 最大角最小: 三角形の内角を大きい順に並べた列が辞書式順序で最小となる三角形分割を作る。

(3) アスペクト比最大: 三角形の内接円半径 r と外接円半径 R の比 r/R を、その三角形のアスペクト比という。三角形のアスペクト比を小さい順に並べた列が辞書式順序で最大となる三角形分割を作る。

(4) 最長辺・高さ比最小: 三角形の 3 辺のうちの最大の長さとその辺に対する高さの比

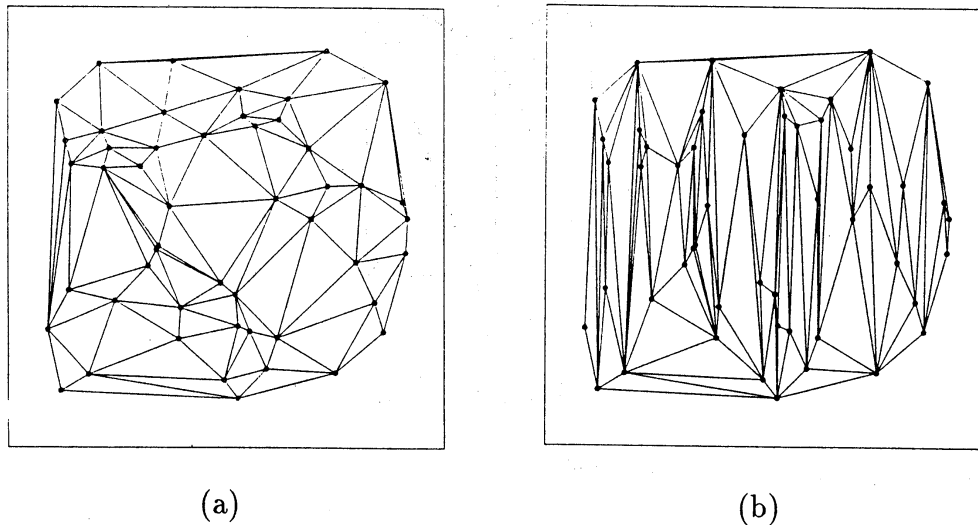


図 2. 同一の頂点集合を用いた 2 種類の三角形分割

を大きい順に並べた列が辞書式順序で最小となる三角形分割を作る.

(5) 重み最小三角形分割: 分割に用いる辺の長さの総和を最小にする三角形分割を作る.

(6) 内接円・包含円比最大: 三角形の内接円の半径とその三角形の最小包含円の半径の比を小さい順に並べた列が最大となる三角形分割を作る.

(7) 最大外接円最小: 三角形の外接円を大きい順に並べた列が辞書式順序で最小となる三角形分割を作る.

(8) 最大最小包含円最小: 三角形の最小包含円を大きい順に並べた列が辞書式順序で最小となる三角形分割を作る.

(9) 最小内接円最大: 三角形の内接円を小さい順に並べた列が辞書式順序で最大となる三角形分割を作る.

これらの基準のうち (1) と (7) を達成するものは一致し, それは点集合 V に対する Delaunay 三角形分割とよばれている [1, 10, 24].

そしてこれは次の方法で作ることができる. 図 3(a), (b) に示すように凸四角形に対角線を入れて三角形に分割する方法は二通りある. 対角線をとりにかえて, これらの一方から他方へ移る操作をフリップという. 図 3 では 6 個の内角のうちの最小値は, (a), (b) それぞれに黒丸で示した角に対応する. そしてその値は (b) の方が大きいから, (b) が基準 (1) を満たす. このことは, (b) に示すように, 三角形の外接円が残りの頂点を内部に含まないとき基準 (1) が満たされると特徴づけることもできる. V を頂点集合とする任意の三角形分割から出発して, その中の凸四角形でそこだけに限って見たとき基準 (1) を満たさないものがあれば——いいかえると, 三角形の六つの内角の最小値がフリップによって大きくなるならば——フリップを行なうということをくり返すと, Delaunay 三角形分割へ収束する [24].

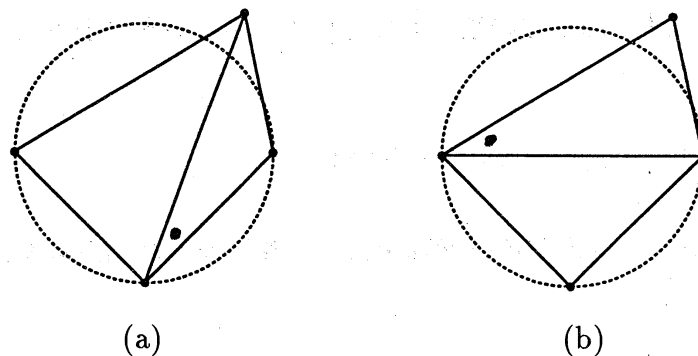


図 3. 対角線のフリップ

頂点の数を n とするとき、この方法では $O(n^2)$ の計算時間が必要となるが、分割統治法などを使うことによって、最悪の場合でも $O(n \log n)$ の計算時間で Delaunay 三角形分割を構成できる [26]. フリップ操作で収束するという性質から、Delaunay 三角形分割は、外接円が他の頂点を含まない三角形を集めてできたものという特徴付けもできる。

しかし、これが、上の諸基準のうち小さい計算量で達成できる唯一の場合である。これ以外の基準を満たそうとすると一般に大きな計算時間が必要となる。たとえば、他の基準がローカルに満たされるようにフリップをくり返す方針も考えられるが、それによって得られるのは極小値または極大値であって、最適なものに到る保証はない。

たとえば、基準 (5) を達成する三角形分割を構成する問題は、NP 完全に属するのか P に属するのかが未だ分かっていない。その観点から理論家の興味を引き、多くの研究がなされている。それらは、目的の三角形分割に必ず使われる辺や絶対に使われない辺をできるだけ多く認識してふるいにかけるという前処理の工夫がほとんどで、それによって少しは早く計算できるようになってきてはいるが、そこで扱うことのできるのは実用的規模のメッシュと比べるとはるかに小さいものとどまっている [6, 7, 9, 19, 30, 32].

このように、形の良いメッシュの基準として多くのものが考えられるにもかかわらず、計算時間の制約から使えるものは Delaunay 三角形分割しかないと言えよう。他の基準を採用する場合には厳密にその基準を満たすメッシュを追求するのではなく、準最適なものがまんするのが実用的である。したがって特に、ローカルな点の配置のみからメッシュ構造を決定する試み [23] (この場合は大域的につじつまを合わせるために、基準を厳密に満たさなければならない) では、Delaunay 三角形分割以外の構造を使うことはほぼ絶望的である [29].

3次元の場合にも良いメッシュを得るための同じような基準が考えられるが、そのどれをとっても厳密な意味で最適なメッシュを作ることは難しい。わずかに望みがあるのは基準 (1) と (7) を3次元に書き直したものである。3次元で (1) に対応する基準としては、次の二つが考えられる。

(1') 最小面角最大: 四面体の面の交角を小さい順に並べた列が辞書式順序で最大となる四面体分割を作る.

(1'') 最小立体角最大: 四面体の頂点での立体角を小さい順に並べた列が辞書式順序で最大となる四面体分割を作る.

一方, 基準 (7) の 3 次元版は次のとおりである.

(7') 最大外接球最小: 四面体の外接球を大きい順に並べた列が辞書式順序で最小となる三角形分割を作る.

2 次元の場合と違って, 基準 (1'), (1''), (7') は等価ではない [27]. (1') または (1'') を満たす四面体分割の効率の良い作り方は知られていない. 一方 (7') を満たす四面体分割は 3 次元 Delaunay 分割とよばれ, 効率よく作ることができる. 実際, 逐次添加法とよばれる技術によって, 最悪の計算量が $O(n^2)$, 平均の計算量がほぼ $O(n)$ のアルゴリズムを作ることができる [14].

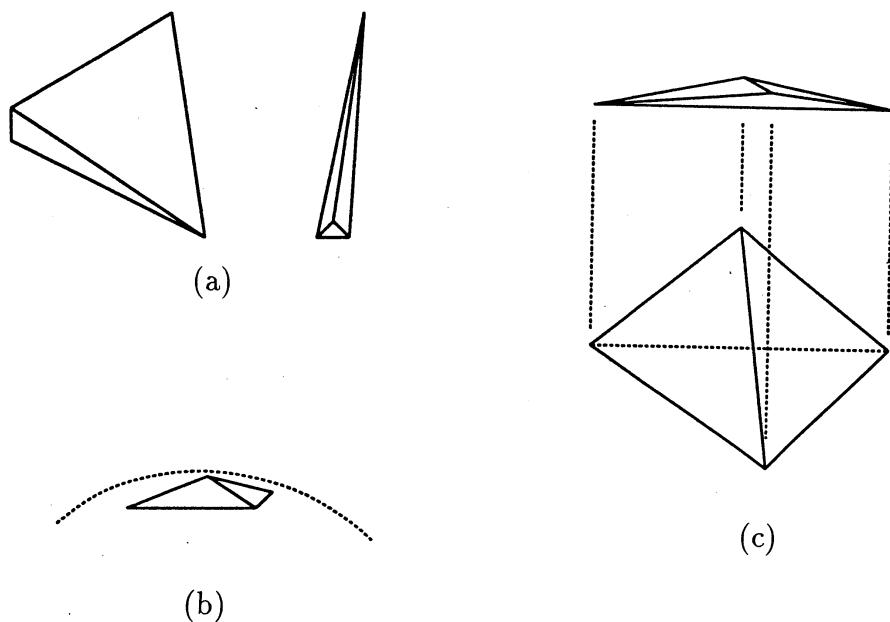


図 4. 好ましくない四面体の分類

しかしながら, 3 次元では (7') を満たす四面体分割が必ずしも良いメッシュを与えないという問題がある. 一つの四面体の外接球の半径を R , 最長辺の長さを L , 最短辺の長さを l としよう. そして $\omega = R/L$, $\kappa = L/l$ とおく. このとき好ましくない四面体は次の 3 種類に分けることができる.

- (i) ω が小さいが κ が大きい場合 (図 4(a)).
- (ii) ω が大きい場合 (図 4(b)).
- (iii) ω も κ もともに小さいが, 体積が 0 に近い場合 (図 4(c)).

3 次元 Delaunay 分割は外接球が他の頂点を含まない四面体から構成される. したがって確かに (i) や (ii) の種類の四面体は生じにくい, (iii) は特に生じにくいわけではない. このように Delaunay 四面体分割は必ずしも良い 3 次元メッシュとなるわけではない.

3.2 メッシュの改良

メッシュ生成のために使う点は, 境界上の頂点を除いて, その数や位置に任意性がある. この任意性を利用して, いったん作ったメッシュをより良いものに改良できる.

2 次元の場合に最も効果を奏する方法の一つは Laplacian スムージングとよばれる処理である. これは, 領域内部の点を, それと隣接するまわりの点の重心へ移動させ, そのあと改めてメッシュを作りなおす操作である. 図 5 に 1 例を示す. この図の (a) は適当に配置した点に対する Delaunay 三角形分割によって作ったメッシュで, (b) は内部のすべての点に Laplacian スムージングを同時に 1 回適用した結果である. この図でも分かるように, この操作で細長い三角形を劇的に減少させることができる.

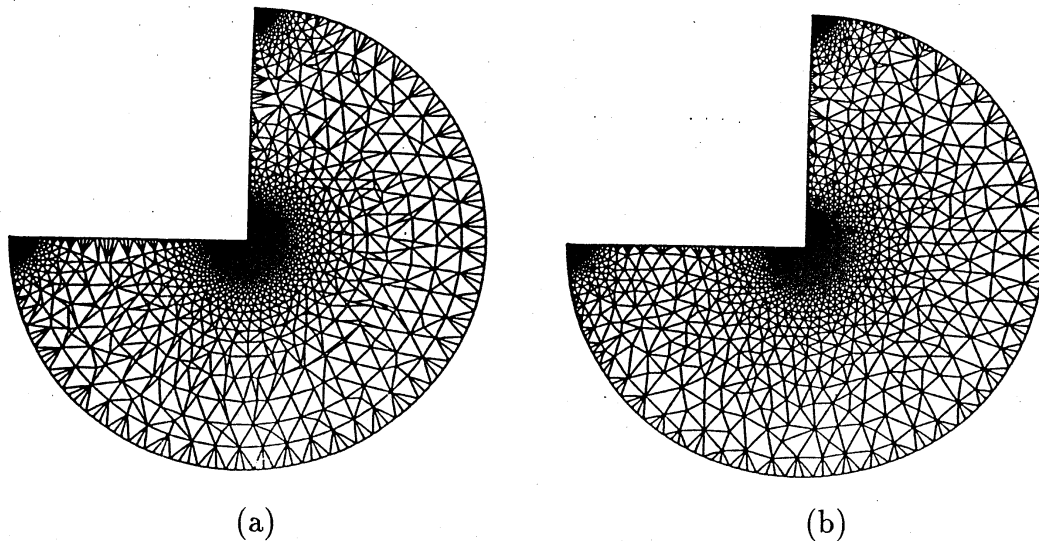
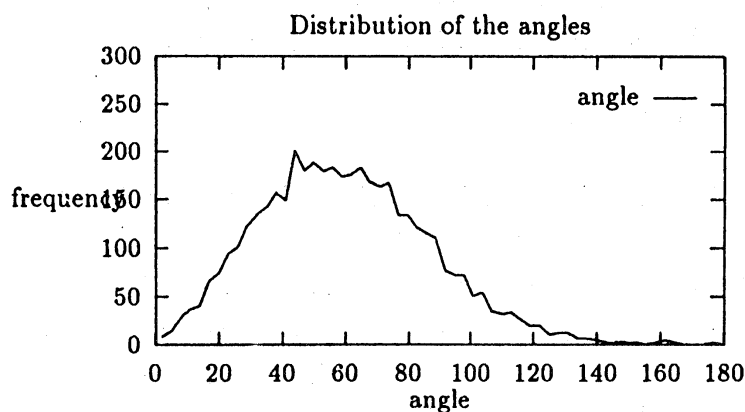


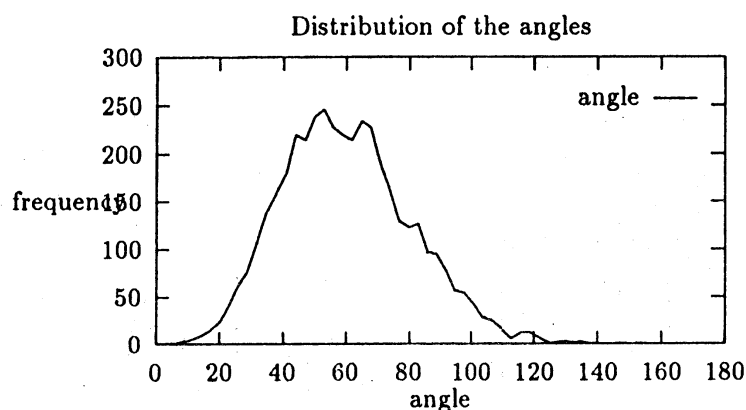
図 5. Laplacian スムージング

一方, 3 次元の場合には状況はもう少し困難である. なぜなら, 図 4(c) に示したタイプの四面体は Laplacian スムージングでも改良されないからである. 図 6 にその 1 例を示す. 図 6(a) は, 立方体の表面と内部にランダムに点を配置して作った Delaunay 四面体分割において, 四面体の境界をなす三角形の内角のヒストグラムを表す. そして図 6(b) は, Laplacian スムージングを 1 回施したあとの同様のヒストグラムである. この図からわかる

ように、三角形の内角は 0 度や 180 度に近いものが減り、60 度付近に分布が集中することがわかる。一方、図 7(a) は、図 6(a) と同じ四面体分割における四面体の隣り合う面同士の交角のヒストグラムで、図 7(b) は図 6(b) と同じ四面体分割に対するものである。この図からわかるように、面の交角に注目すると、Laplacian スムージングを施しても、0 度付近や 180 度付近のものが残る。この傾向は Laplacian スムージングを何回施しても変わらない。



(a)



(b)

図 6. Laplacian スムージングによる三角形の内角の変化（文献 [15] より）

このように 3 次元においては、Delaunay 四面体分割の単純な適用も、Laplacian スムージングによる変更も、必ずしも良質のメッシュを生み出すわけではない。そのために、個々の局所的メッシュ形状に沿った個別の変形ルール [16, 18, 22], 点の配置と四面体生成を並行して行なう方法 [20, 31, 12, 5] など、多くのヒューリスティックな手法が提案されているが、すべての場合にうまくいく方法は未だ得られていないのが現状である。

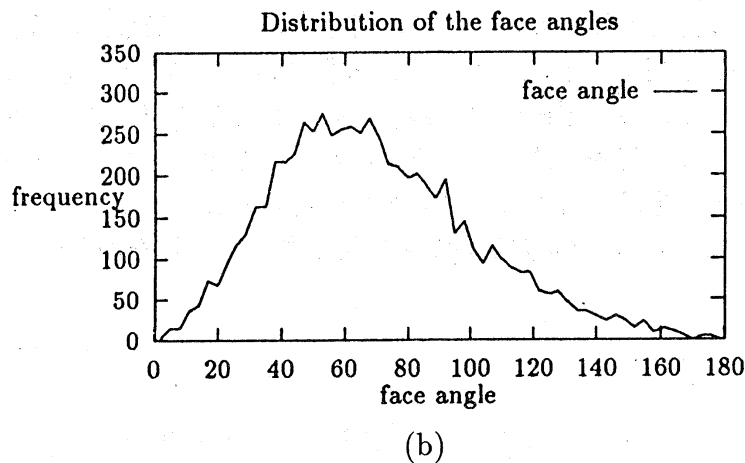
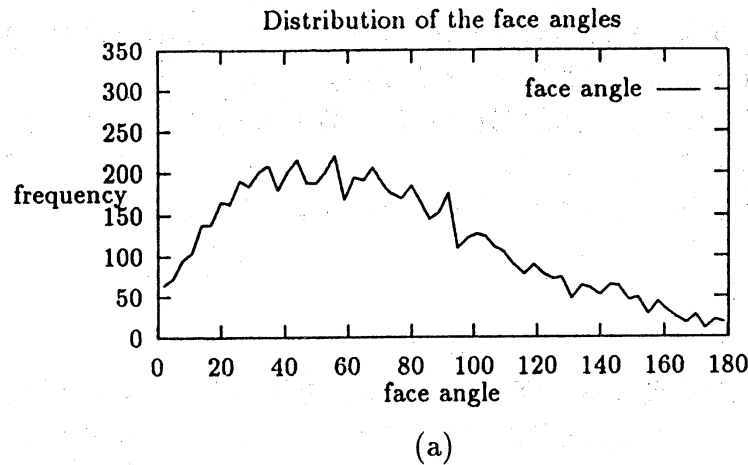


図 7. Laplacian スムージングによる面の交角の変化 (文献 [15] より)

4 境界との両立性

メッシュは与えられた領域の内部を要素に分割する方法である。一方、点集合に対する Delaunay 分割は、その凸包の内部の分割であるから、必ずしも与えられた領域の境界と両立するとは限らない。したがって、領域の境界と両立するメッシュを作る技術も必要である。

この技術は、大きく二つに分けることができる。その第一は、Delaunay 分割が境界の形と両立するようにあらかじめ点の数と配置を注意深く選ぶ方法である。Delaunay 四面体分割は、外接球が他の点を含まない四面体から構成される。そこで、「領域の境界には十分高い密度で点を配置し、領域内部の点は境界から十分離す」という方針によって境界の内部と外部にまたがる四面体の発生を防ぐことができる [8]。また一度 Delaunay 四面体分割を作ってから、境界を貫く四面体があったらその近くの境界上に新しい点を追加して Delaunay 四面体分割を作りなおすという方針も有効である [17, 18]。

もう一つの技術は、Delaunay 分割の代わりに、境界を考慮して Delaunay 分割を変更した構造を作る方法である。すなわち、境界をなす三角形面の集合を制約とみなし、(i) 制約の両側にまたがることのない 4 点で、(ii) それらを通る球から制約の反対側を除いた部分

に他の点が含まれない，という性質を満たすものに対して四面体を作る．このようにしてできるすべての四面体の集合は領域内部の分割を与える．この分割は，境界三角形面の集合を制約とする制約付き Delaunay 四面体分割とよばれる [3, 5].

いずれの技術を適用しても，前節で見たとおり Delaunay 分割では十分良質のメッシュが得られるわけではないので，さらに改良しなければならない．この改良の際にも，境界の両側にまたがる四面体を作らないように注意しなければならない．

5 おわりに

偏微分方程式の離散化に必要な 3 次元単体メッシュを作る技術の現状と残されている課題について概観してきた．ここで見てきた課題のほかに，メッシュ生成アルゴリズムの数値的安定化という課題も残っている．すなわちメッシュ生成の計算の途中で数値誤差が発生するため，アルゴリズムが理論どおりに動作するとは限らない．この数値的不安定性という観点からも 3 次元 Delaunay 分割は 2 次元に比べて，格段に難しいことが指摘されている [28]. しかしこれはメッシュ生成に限らず幾何情報処理全般にわたる問題である．幸いこの数値的安定化をはかるために，「位相優先法」，「記号摂動法」などのソフトウェア設計方法が提案されており，メッシュの生成にもこれらの方法を援用することができる．

なお本調査研究は，文部省科学研究費補助金の援助のもとで行なわれたものである．

References

- [1] F. Aurenhammer: Voronoi diagrams — A survey of a fundamental geometric data structure. *ACM Computing Survey*, vol. 23 (1991), pp. 345–405.
- [2] I. Babuska and A. K. Aziz: On the angle condition in finite element method. *SIAM Journal on Numerical Analysis*, vol. 13 (1976), pp. 214–226.
- [3] T. Baker: Automatic mesh generation for complex 3-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers*, vol. 5, nos. 3–4 (1989), pp. 161–175.
- [4] M. Bern and D. Eppstein: Polynomial-size nonobtuse triangulation of polygons. *Proceedings of the 7th Annual Symposium on Computational Geometry*, North Conway, pp. 342–350, 1991.
- [5] M. Bern and D. Eppstein: Mesh generation and optimal triangulation. In F. K. Hwang and D-Z. Du (eds.), *Computing in Euclidean Geometry*, World Scientific, pp. 23–90, 1992.
- [6] S. W. Cheng and Y. F. Xu: Approaching the largest β -skeleton within a minimum weight triangulation. *Proceedings of the 12th Annual Symposium on Computational Geometry*, Philadelphia, 1996, pp. 196–203.

- [7] S. W. Cheng, N. Kato and M. Sugai: A study of the LMT-skeleton. T. Asano *et al.* (eds.) *Algorithms and Computations — 7th Int. Symp., ISAAC'96* (Lecture Notes in Computer Science, 1178), Springer-Verlag, Berlin, pp. 256–265.
- [8] T. Dey, C. L. Bajaj and K. Sugihara: On good triangulations in three dimensions. *International Journal of Computational Geometry and Applications*, vol. 2 (1992), pp. 75–95.
- [9] M. T. Dickerson and M. H. Montague: The exact minimum weight triangulations. *Proceedings of the 12th Annual Symposium on Computational Geometry*, Philadelphia, 1996, pp. 204–213.
- [10] H. Edelsbrunner: *Algorithms in Combinatorial Geometry*, Springer-Verlag, New York, 1987.
- [11] D. A. Field: The legacy of automatic mesh generation from solid modeling, *Computer Aided Geometric Design*, vol. 12 (1995), pp. 651–673.
- [12] P. L. George: *Automatic Mesh Generation. Applications to Finite Element Methods*. Chichester, U. K., Wiley, 1991.
- [13] K. Ho-Le: Finite element mesh generation methods: a review and classification. *Computer Aided Design*, vol. 20 (1988), pp. 27–38.
- [14] 稲垣, 杉原: 3次元ドロネー図の構築における退化に起因する問題点とその対策. 電子情報通信学会論文誌, D-II, vol. J79-D-II (1996), pp. 1696–1703.
- [15] 稲垣宏: 数値的に安定な 3 次元 Voronoi 図構成算法とその応用. 学位請求論文, 東京大学大学院工学系研究科計数工学専攻, 1996.
- [16] B. Joe: Three dimensional triangulations from local transformations. *SIAM Journal on Scientific and Statistical Computing*, vol. 10 (1989), pp. 718–741.
- [17] B. Joe: Delaunay versus max-min solid angle triangulations for three-dimensional mesh generation. *International Journal for Numerical Methods in Engineering*, vol. 31 (1991), pp. 987–997.
- [18] B. Joe: Construction of three-dimensional improved quality triangulations using local transformations. *SIAM Journal on Scientific Computing*, vol. 16, no. 6 (1995), pp. 1292–1307.
- [19] J. M. Keil: Computing a subgraph of the minimum weight triangulation. *Computational Geometry: Theory and Applications*, vol. 4 (1994), pp. 13–26.

- [20] A. Liu and B. Joe: Quality local refinement of tetrahedral meshes based on bisection. *SIAM Journal on Scientific Computing*, vol. 6 (1995), pp. 1269–1291.
- [21] D. J. Mavriplis: Adaptive mesh generation for viscous flows using Delaunay triangulation. *Journal of Computational Physics*, vol. 90, no. 2 (1990), pp. 271–291.
- [22] 宮野, 杉原: ドロネー四面体メッシュの品質改良のための一手法. 電子情報通信学会技術報告, COMP96-76 (1996).
- [23] 長嶋, 他: 節点単位 of の処理に基づくメッシュレス法の開発. 日本機械学会論文集 (A) 62 巻 (1996), pp. 2374–2480.
- [24] A. Okabe, B. Boots and K. Sugihara: *Spatial Tessellations — Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, Chichester, 1992.
- [25] M. A. K. Posenau: Approaches to high aspect ratio triangulation. *Proceedings of the 5th Canadian Conference on Computational Geometry*, 1993, pp. 30–35.
- [26] F. P. Preparata and M. I. Shamos: *Computational Geometry — An Introduction*. Springer-Verlag, New York, 1995.
- [27] V. T. Rajan: Optimality of the Delaunay triangulation in R^d . *Proceedings of the Seventh Annual ACM Symposium on Computational Geometry*, pp. 357–363, 1991.
- [28] K. Sugihara and H. Inagaki: Why is the 3d Delaunay triangulation difficult to construct? *Information Processing Letters*, vol. 54 (1995), pp. 275–280.
- [29] 杉原厚吉: ローカルメッシュでできること——メッシュレス解法は新解法となり得るか. 日本応用数理学会 1998 年度年会講演予稿集, pp. 146–147, 1998.
- [30] B. Yang: A better subgraph of the minimum weight triangulation. *Proc. International Conference on Computing and Combinatorics*, pp. 452–455, 1995.
- [31] M. M. F. Yuen, S. T. Tan and K. Y. Hung: A hierarchical approach to finite element mesh generation. *Internat. J. Numer. Methods Engrg.*, vol. 32 (1991), pp. 501–525.
- [32] B. Yang, Y. Xu and Z. You: A chain decomposition algorithm for the proof of a property on minimum weight triangulation. *Proc. International Symposium on Algorithms and Computation*, pp. 423–427, 1994.