

Voronoi 図構成における 分割統治型算法の誤差対策

大石 泰章, 杉原 厚吉
東京大学工学部計数工学科

Katajainen & Koppinenの研究によって理論的にも実用的にも最も速いVoronoi図構成算法となる可能性を持つに至った分割統治型算法を, 計算の途中で数値誤差が混入しても破綻しないように改良した. 理論的に正しさが保証された算法であっても, 数値誤差の生じる現実の計算機では破綻することがある. この破綻を防ぐために, 数値計算結果が対象の位相的性質と矛盾しそうになったら処理の方向を転換して矛盾を防ぐという形の歯止めを算法の中に挿入する, というのが改良の基本的な方針である. 新しい算法に基づいて作ったプログラムはどんなに低精度で計算しても破綻せず, いままで破綻してしまって求められなかったような種類のVoronoi図も正しく求めることができた. また歯止めをかけても計算量が悪化しないことも確かめられた.

A METHOD FOR AVOIDING NUMERICAL ERRORS IN THE DIVIDE-AND-CONQUER ALGORITHM FOR VORONOI DIAGRAM CONSTRUCTION

Yasuaki OOISHI and Kokichi SUGIHARA
Department of Mathematical Engineering and Information Physics
Faculty of Engineering, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113 Japan

The paper presents a method to make the divide-and-conquer algorithm for constructing Voronoi diagrams to be numerically stable. Because of recent improvement by Katajainen and Koppinen, the divide-and-conquer algorithm attained the lowest time complexity both in the worst-case sense and in the average-case sense. We make this algorithm numerically stable in the sense that the algorithm will never fail however large errors may take place in the course of computation. Our basic idea is that we place higher priority on some topological conditions than on numerical values and thus avoid inconsistency. A computer program based on our new algorithm turned out to have as good features as expected both from a numerical stability point of view and from a time complexity point of view.

1 はじめに

Voronoi図はEuclid平面上に存在する有限個の点に対してその勢力圏を表す図で、多くの応用分野を持つ重要な幾何学的概念である[2]。

Voronoi図の構成算法は逐次添加法と分割統治法の二算法が有力である[2, 6]。これまでは点の数が N 個のとき平均的な計算量が $O(N)$ になる逐次添加法が最も実用的な算法であるとされてきた[6, 7]。しかし最近Katajainen & Koppinenによって分割統治法が改良され、従来 $O(N \log N)$ だった平均的計算量[7]が逐次添加法と同程度の $O(N)$ に引き下げられた[4]。もともと最悪の場合の計算量では分割統治法の方が優れていたため[5]、逐次添加法をしのごく可能性もあるとして今や分割統治法はその魅力を増している。

但し以上の話は「計算の途中では数値誤差が混入しない」という仮定をおいた理想の世界の話である。これに対して実際に計算機上で算法を実行するときには数値誤差の混入がまぬがれ得ないため、理論どおりにVoronoi図が求められるわけではない。実際、経験的に分割統治法では単精度計算で1000点、倍精度計算でも3000点が暴走しないで結果を求められる限界だと言われてきたようである[9]。

杉原・伊理はこの問題を根本的に解決するために逐次添加型のVoronoi図構成算法を改良し、「どんなに計算精度が低くても破綻せず、正しいVoronoi図が満たすべき位相的性質を少なくとも満足する結果を出力し、計算精度を上げていくにつれて出力が正しいVoronoi図に『収束』していく」という新しい算法を構成した。この新しい算法は数値誤差が混入することを前提としているため、退化が生じていても単に退化に近い状態として扱うことができ、例外処理が不要になるという副産物も得られる[3, 9]。ところで杉原・伊理が逐次添加法を取り上げたのは、当時この算法が最も実用的とされていたからである。一方、上で述べたように分割統治法が理論面で改良されてきたために、分割統治法に対して同様の数値的安定化が可能か否かを確かめることは大きな意義を持つに至っている。

本論文では以上の動機に基づいて分割統治型Voronoi図構成算法を上の意味で数値的に安定にする。まず第2章でVoronoi図の定義を与るとともに杉原・伊理の数値的安定化の方法を扱いやすい形に解釈しなおす。その解釈に基づいて第3章と第4章で分割統治型Voronoi図構成算法の数値的安定化を行う。第5章では安定化された算法に従って作ったプログラムが期待された様々の良い性質を満足することを実験によって確かめる。

2 Voronoi図と数値的安定化

2.1 Voronoi図

Euclid平面 R^2 上に指定された N 個の点(これを母点

という)を P_1, P_2, \dots, P_N とする。平面上の2点 P と Q の間のEuclid距離を $d(P, Q)$ と書くとき、母点 P_i のVoronoi領域 $V(P_i)$ は、

$$V(P_i) = \{P \in R^2 \mid d(P, P_i) < d(P, P_j) \text{ for } j=1, 2, \dots, N; j \neq i\} \quad (\text{式1})$$

と定義される。 $V(P_1), V(P_2), \dots, V(P_N)$ がもたらす平面の分割をVoronoi図という。Voronoi図の例を図2.1(a)に示す。ここでそのVoronoi領域が隣合うような二つの母点を線分でつなぐことにすれば、Voronoi図の双対図形が得られる。これをDelaunay網といい、つないだ線分をDelaunay辺という(図2.1(b))。

定義から直ちに次の性質が導かれる。

(P1) Delaunay網は母点集合の凸包内部の凸多角形分割である。

また、この凸多角形は一般には三角形であるが、4個以上の母点が同一円周上に載ると、三角形以外の凸多角形を生じることがある(この状態は退化と呼ばれる)。以下では、このような多角形は任意の対角線を導入して三角形に分割するものとする。したがって、Delaunay網は母点集合の凸包内の三角形分割である。このとき次の性質も成り立つ。

(P2) 3個の母点 P_i, P_j, P_k を頂点とする三角形がDelaunay網の三角形と一致するのは、 P_i, P_j, P_k を通る円の内部に他の母点が存在しないときに限る。

2.2 数値的安定化の基本方針

Voronoi図構成に限らずある幾何的アルゴリズムを前章の意味で数値的に安定にする方法は、次のように解釈できると思われる。即ちこうした算法の多くは扱う図形の位相構造をグラフなどの組合せ論的方法で表現し、図形の計量的性質に関する浮動小数点計算の結果

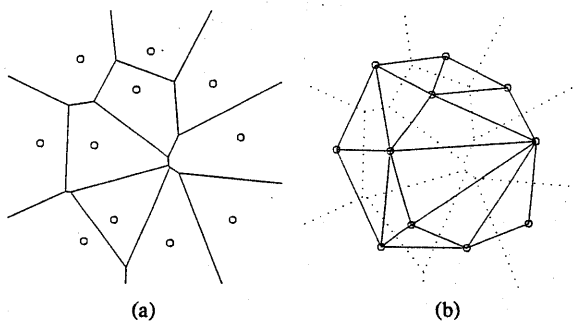


図2.1 Voronoi図とDelaunay網(小丸は母点の位置を表す): (a)Voronoi図; (b)Delaunay網。

にしたがって、この位相構造を変更していくことで構成されている。ここで数値誤差の混入で算法が破綻するのは、誤差のために浮動小数点計算が誤差分だけずれた結果を出し、そのために誤った分岐をして誤った位相構造変更を行ってしまうからである。だから破綻を防ぐためには、数値計算が正しければ行われるはずのない矛盾した位相構造変更がなされないか否かを監視し、矛盾が発生しそうになったら浮動小数点計算の結果より位相構造の一貫性を優先することによってその矛盾を解消すればよい。つまり、最低限算法が破綻しないだけの位相的性質を保証するための監視手続きを、矛盾に陥らないための歯止めとして算法に挿入すればよい。そのような位相的な歯止めは次の性質を満たすものでなくてはならない。

- (Q1) 算法が正しく動いている限り、決してその動きを妨げないような歯止めであること。
- (Q2) その歯止めをかけておけば（どれほど大きな数値誤差が発生しても）算法が必ず有限回の動作の後に結果を出力して終了すること。
- (Q3) 算法が正しく動いている場合は、数値的安定化をする前の算法と比べて計算量が著しく悪化しないこと。
- (Q4) (Q1)～(Q3)に反しない限り、できるだけ厳しい歯止めであること。

位相的性質の監視や位相構造の変更は組合せ論的演算のみによって行われるので数値誤差の心配がない。したがって、上の性質を満たす位相的歯止めによって矛盾の発生が完全に防止できることに注意されたい。これで前章の意味の数値的安定化が実現される。

3 マージの算法の数値的安定化

3.1 分割統治法を数値的に安定にするには

本論文ではDelaunay網の言葉を用いて算法を記述する。その理由はこちらの方が数値的安定化が容易であると思われるからである[8]。

Katajainenらによる改良型でも、従来型でも、分割統治法では、まず母点集合を十分小さな部分集合に分割してそれぞれに対するDelaunay網を作り、次に二つのDelaunay網を融合させて、一つのDelaunay網にする手続き（この手続きを以下ではマージという）を繰り返す[1, 5]。但し改良型では、従来型と違って水平方向だけでなく垂直方向のマージも行なう[4]。さて分割統治法を数値的に安定にするためにはマージの算法に上で述べた条件を満たす歯止めをかけてやればよい。この方法を本章と次章で述べる。本章ではマージの主要部分について考え、次章ではその前処理である共通接線発見法について考える。本章と次章で紹介する歯止めをかけることで分割統治法は完全に数値的に安定と

なる。

3.2 マージの算法[1, 5]

二つのDelaunay網FとSが与えられたとき、これらをマージして一つのDelaunay網を作ることを考える。但し、Fに属す全ての母点の x 座標はSに属す全ての母点の x 座標より小さいものとする。この場合のマージを水平方向のマージという。Katajainenらの算法ではFに属す全ての母点の y 座標がSに属す全ての母点の y 座標より大きい場合のマージ（これを垂直方向のマージという）も必要になるが、これは以下の算法を90°回転させて行うことにより実現できる。

まずFとSの下側共通接線と上側共通接線を求める（正確にはFに属す母点の凸包とSに属す母点の下側からの共通接線等というべきであるが、誤解のおそれのない限りこのような簡潔な表現を用いることにする）。FとSのマージはFとSとを結ぶDelaunay辺（以下これをF-Sの辺と呼ぶ。同様に両端点が共にFに[Sに]属するDelaunay辺をF-Fの辺[S-Sの辺]と呼ぶ）を引いていくことで行われるが、これを下から上へ順に引いて行く方法を考える。性質(P1)より、一番はじめのF-Sの辺は下側共通接線に沿って引かれ、最後のF-Sの辺は上側共通接線に沿って引かれることがわかる。F-Sの辺を1本引いた直後から次のF-Sの辺を引くまでの過程を図3.1に示す。但し、一点鎖線はF-Fの辺を、破線はS-Sの辺を、実線はF-Sの辺を表す。(a)はF-Sの辺を1本引いた直後の状態である。一番最近に引かれたF

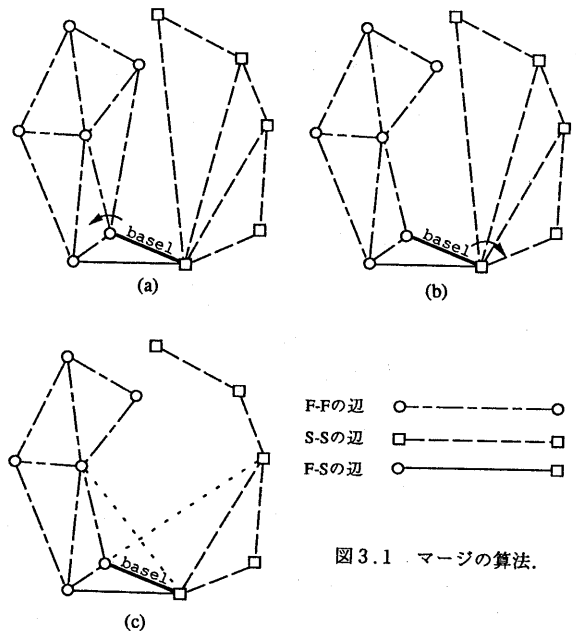


図3.1 マージの算法。

-Sの辺をbaselと呼ぶことにする。算法では、まずbaselのF側端点を要として反時計回りにbaselの次の辺から順にF-Fの辺を調べ、マージ後のDelaunay辺とはならないF-Fの辺を削る。この例では(b)に示すように1本の辺が削られる。次にbaselのS側端点を要として時計回りにbaselの次の辺から順にS-Sの辺を調べ、マージ後のDelaunay辺とはならないS-Sの辺を削る。この例では(c)に示すように2本の辺が削られる。最後に(c)に点線で示した二つの引き方のうちどちらか一方を選んで新しいF-Sの辺とし、これが新たなbaselになる。以上の操作を繰り返してマージは行われる。

ここでどの辺まで削除すべきであるかおよび、どちらの引き方を選ぶべきかという判断は、性質(P2)に基づいて行なう。母点A, B, C, Dの座標を(x_i, y_i) (i=A, B, C, D) とするとき、A, B, Cを通る円の内部にDが含まれるか否かは、

$$\begin{vmatrix} x_A & y_A & x_A^2 + y_A^2 & 1 \\ x_B & y_B & x_B^2 + y_B^2 & 1 \\ x_C & y_C & x_C^2 + y_C^2 & 1 \\ x_D & y_D & x_D^2 + y_D^2 & 1 \end{vmatrix} \quad (式2)$$

の符号によって判断することができる(煩雑になるのでこれ以上述べないが詳しくは文献[1,5]を参照されたい)。(式2)の値は数値計算によって行われるため、誤差の生じる恐れがある。従って数値的安定化が必要となる。

3.3 マージの算法の数値的安定化

本論文では次の三つの歯止めを提案する。

(C1) 『連結性条件』：それを削除すると、グラフ構造が非連結になってしまうF-Fの辺またはS-Sの辺は削ってはいけない(図3.2)。

(C2) 『横断性条件』：その両端点が共に、既に引かれたF-Sの辺に共有されているF-Fの辺またはS-Sの辺は削ってはいけない(図3.3(a))。

(C3) 『並列枝禁止条件』：それを付加すると、既に存在している他の辺の並列枝となってしまうような辺は引いてはいけない(図3.4(a))。

この歯止め(C1)～(C3)が、歯止めとしての条件(Q1)～(Q3)を満たすことを以下に示す。

まず(Q1)を考えよう。正しいDelaunay網は連結であり並列枝を持たないので、計算が正しく行われている限り(C1)『連結性条件』と(C3)『並列枝禁止条件』が算法の動きを妨げることはない。また(C2)『横断性条件』を満たさないような辺の削除を行うと、前節の算法に従えばいつかはF-Sの辺

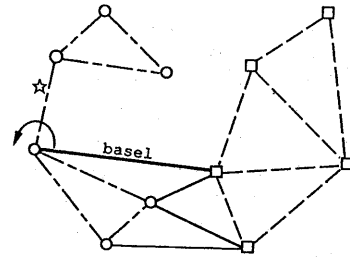


図3.2 連結性条件: ☆印の辺を削ってはいけない。

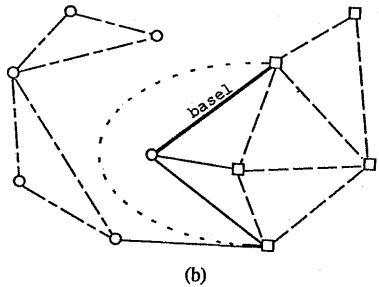
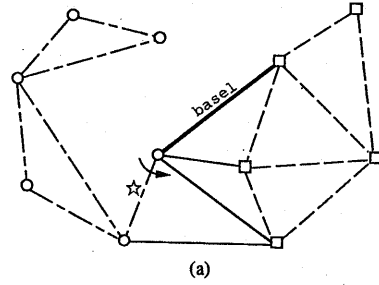


図3.3 横断性条件: ☆印の辺を削ってはいけない。

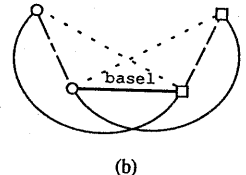
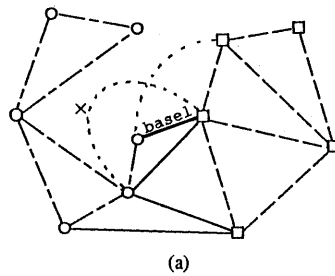


図3.4 並列枝禁止条件: ×印の辺を付加してはいけない。

(横断辺)でない辺が付加されてしまう。例えば図3.3(a)の☆印の辺を削ってしまうと、(b)のようにいつかはS-Sの辺を結ぶことになってしまうが、これはS中の母点に対するDelaunay網が既にできあがっていたことに反する。よって計算が正しく行われている限

り (C2) が算法の動きを妨げることはない、これで歯止めとしての条件 (Q1) が満足されることが示された。

次に (Q2) を考える。前節の算法の定義から、F と S が平面グラフならマージの結果も平面グラフとなることが保証される。したがって、二つの辺が交差することはないから、(C3) を課したせいで図 3.1 (c) の二つの引き方のどちらも選べなくなって算法が行き詰まることはない (図 3.4 (b))。F と S に属す母点の数は有限であるから、各 base1 の端点を要とする辺の除去作業は有限回しか行われない。また算法中でいったん引いた F-S の辺は、その回のマージが終了するまでは削られることはないから、F-S の辺を引く作業も有限回しか行われない。条件 (C1) より、最初の F-S の辺を引いた後は全ての母点为一个の連結グラフに属すから、有限回の作業の後に、少なくとも上側共通接線と一つの端点を共有する F-S の辺を引くことが保証される。一般性を失うことなく、図 3.5 に示すように、上側共通接線と S 側端点を共有する F-S の辺を引いたとできる。この後、新しく引く F-S の辺の S 側端点は動かさないから、新しい F-S の辺の引き方は図 3.1 (c) のように二通りではなく、図 3.5 のように一通りしかない。このただ一つの候補辺が (C3) 『並列枝禁止条件』に反するときにはこれでマージを終了し、反しない場合にはさらに作業を続けることにすれば、マージは必ず上側共通接線に沿った辺を引くところまで行き着いて終了するか、あるいは条件 (C3) に反したところで終了することになる。したがって、歯止めの条件 (Q2) を満たすことができる。

(Q3) を調べよう。(C1)、(C2) の歯止めは、簡単に言ってしまうと F または S の外周の辺はある条件を満たすときには削れないということである。よって F と S の外周の辺にその条件をうまく表すような印をつけてやれば、辺を削ろうとする度に印を調べることで計算量を増加させることなく (C1)、(C2) を組み込むことができる (詳しくは文献 [8])。また、並列枝が生じないかどうかは、引こうとする辺の端点の一つから出る辺を全て調べてやればわかるから、

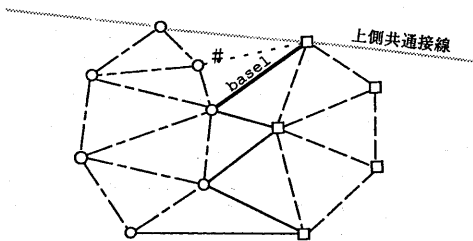


図 3.5 マージ終了直前の様子 (#印のようにしか付加できない)。

平均的には引く辺 1 本当たり定数時間ですむ。よって計算が正しく行われている限り (C1) ~ (C3) の歯止めを組み込んでも計算量のオーダーは増加しない。即ち歯止めの条件 (Q3) も満たされる。

平面グラフが Delaunay 網と同型であるための組合せ論的な必要十分条件が明らかになっていない現状では、条件 (Q4) は証明できるという性質のものではない。さらに強力な歯止めがかけられるか否かは今後の検討課題である。

4 共通接線発見法の数値的安定化

4.1 共通接線発見法 [6]

マージの算法において必要な共通接線の発見法について説明する。以下では、水平方向のマージにおける下側共通接線の発見法を例にとることにする。他の共通接線の求め方も方向が変わるだけで本質的に同じである。前章と同様に、マージする二つの Delaunay 網のうち左側のものを F、右側のものを S と呼ぶ。

まず、F と S の外周上の母点から適当なものを 1 点ずつ選んで、それぞれ ST_1 、 ST_2 とする。この 2 点を結ぶ直線が最初の共通接線候補となる。さて S の外周上で ST_2 の両隣りにある 2 点を調べ、現在の共通接線候補より下にあるものがないかみる。もしそうした母点があればそれを新しい ST_2 として共通接線候補を更新する。同じように F の外周上で ST_1 の両隣りにある 2 点を調べ、現在の共通接線候補より下にあるものがあれば、その母点を新しい ST_1 として共通接線候補を更新する。以下この操作を繰り返していけば、 ST_1 も ST_2 も変更されなくなった時点で下側共通接線が求まる。ある点 $A(x_a, y_a)$ が $ST_1(x_1, y_1)$ と $ST_2(x_2, y_2)$ とを結ぶ共通接線候補より下にあるということは、

$$\begin{vmatrix} x_2 & y_2 & 1 \\ x_1 & y_1 & 1 \\ x_a & y_a & 1 \end{vmatrix} > 0 \quad (\text{式 3})$$

と同値であるから [1]、これを用いて判定できる。但し、(式 3) の符号判定は数値計算を要するから、共通接線発見法にも数値的安定化が必要であり、位相的歯止めをかけなくてはならない。

4.2 共通接線発見法の数値的安定化

まず、いくつか言葉を定義しておく。まず左側の Delaunay 網 F の中で x 座標が最大 [最小] の母点 (最大 [最小] のものが複数あるときにはその中で y 座標が最大 [最小] のもの。以下同様) を R_1 [L_1] と呼び、 y 座標が最大 [最小] の母点を T_1 [B_1] と呼ぶ。右側の Delaunay 網 S の中の母点についても同様に R_2 、 L_2 、 T_2 、 B_2 を定める。

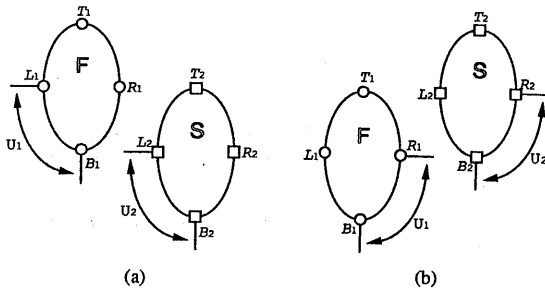


図 4.1 下側共通接線の端点の存在範囲：
(a) B_1 の方が B_2 より高いとき；
(b) B_2 の方が B_1 より高いとき。

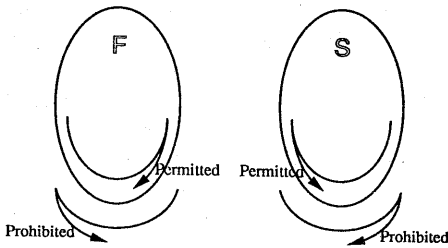


図 4.2 下側共通接線の端点探索における動き方の制限。

さて位相的歯止めとして次のものを考える。

(C4) 共通接線の端点の探索は図 4.1 に示す範囲 U_1, U_2 の中で行う。

(C5) ST_1 [ST_2] を時計 [反時計] 回りに動かした後は, (ST_2 [ST_1]) を動かして状況が変化しても ST_1 [ST_2] を反時計 [時計] 回りに戻りさせて動かしてはいけない (図 4.2)。

歯止め (C4) は B_1, L_1 など座標の最大・最小で定まる母点を用いているから, これを「位相的」と呼ぶのは無理があろう。しかし二数の大小比較は計算誤差の混入無しに行えるから, 「完全に信頼できるもの」という点で位相的歯止めと準ずるものとして扱うことにする。

上の二つの歯止めが歯止めの満たすべき条件 (Q1) ~ (Q3) を満足することを示す。

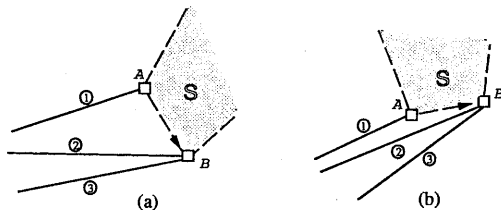


図 4.3 歯止め(C5)をかけられる理由。

まず (Q1) を調べる。計算が正しく行われているとき F の外周も S の外周も凸だから (C4) が満たされることは明らかであろう。(C5) は S 側端点 S_{T_2} の場合を例にして説明する。図 4.3 で ST_2 は (C4) の制限により S の下半分しか動かず, 計算が正しければ S の外周は凸なので, ST_2 が反時計回りに一つ隣の母点に移るとき, 移る方向は (a) のように右下か, (b) のように右上か二通りしかない。どちらの場合も ST_2 が母点 A から B へと移った後 (①→②), ST_1 の移動 (②→③) があっても, そのとき ST_1 は ST_2 の回りを反時計回りに移動するから, 母点 A が ST_1 と ST_2 を結ぶ直線より上方にあるという性質は保たれる。したがって, ST_2 を B から A へと時計回りに戻す必要が生じることはない。

次に (Q2), (Q3) を調べる。歯止め (C5) をかけたため, ST_1 と ST_2 はたかだか (C4) で制限された存在範囲 (図 4.1 の U_1, U_2) を一往復するだけである。よって有限時間で算法は終了し, (Q2) が満たされる。また (C4), (C5) は, 特に工夫しなくても計算量を増加させることなく組み込むことができる。即ち (Q3) が満足されている。

共通接線候補の初期値は, (C4) を満たす範囲でどのように選んでもよい。本論文では B_1 と B_2 をそれぞれ ST_1 と ST_2 の初期値とした。母点がランダムに分布するとき下側共通接線が一番近そうなのが, 母点 B_1 と B_2 を結ぶ直線だと思われるからである。

5 実験

3章と4章で述べた方針に従って算法を構成し, それに基づいてプログラム (C言語で合計約2700行) を作成して数多くの計算実験を行った。その結果どんなに低い精度で計算しても決して暴走することなく計算結果を出力して終了し, 計算精度を上げるにつれてその出力結果が正しい Voronoi 図に収束していくことが確認された。また計算量の面でも良い性質を持つことが確認された。なお実験はすべて東京大学教育用計算機センター富士通 FACOM380 の UNIX System V 上で行った。

図 5.1 は本プログラムの計算時間の実測結果をまとめたものである。領域 $[0, 1) \times [0, 1)$ 中にランダムに分布する 2^7 個, 2^8 個, ..., 2^{16} 個の母点に対して各母点数で 10 回ずつ測定して平均をとった。横軸は母点数の対数を表し, 縦軸は母点一個当たりの平均時間を表す。本プログラムは Katajainen ら以前の従来型の分割統治法でも, Katajainen らによる改良型の分割統治法でも Voronoi 図構成ができるが, どちらの方法を用いた場合にも数値的安定化によって計算複雑度が悪化していないことがわかる。即ち従来型で $O(N \log N)$, 改良型でほぼ $O(N)$ が達成されている。

図 5.2 は計算精度を上げるにつれて, プログラムの出力結果が正しい Voronoi 図に「収束」していく様子を

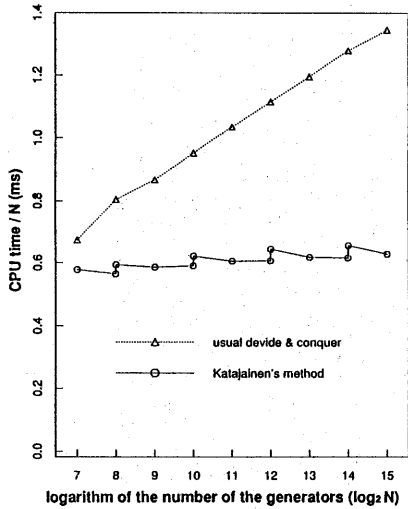


図 5.1 計算時間の比較.

示したものである。与えた母点は $[0, 1) \times [0, 1)$ 中に一様乱数を用いて発生させた30点である。ここで低精度計算のシミュレートは、浮動小数点計算を構成する一つ一つの二項演算が行われるごとに、その結果の仮数を2進数で表示したものの上から $n+2$ 桁目を0捨1入して、 $n+1$ 桁に丸めることによって行った。例えば $n=5$ は浮動小数点計算を仮数表現桁数2進6桁の精度で行ったことを表し、特に $n=23$ は IEEE の単精度の規格に一致する。

図 5.3 は $[0, 1) \times [0, 1)$ 中にランダムに分布する4000点に対して単精度計算 ($n=23$) で求めた Voronoi 図の左上の四分の一である。正しく構成できているように見える。かつて単精度で構成できる限界が1000点だったのと比べると格段の性能の向上であり、しかも感

触ではさらに多くの点に対して正しく構成できそうである(この母点数までしか実験しなかったのは1ユーザ当りの記憶容量制限のためである)。また同一円周上に分布する2000点についても単精度計算で Voronoi 図を構成した。このような母点配置は非常に Voronoi 図を構成しにくい例として知られているが、本プログラムは破綻することなく終了し、通常の倍率で見限り正しい Voronoi 図(正しい Voronoi 図の近似図形)を出力した。

6 おわりに

最速の Voronoi 図構成算法となる可能性を持つようになった分割統治法を杉原・伊理の言う意味で数値的に安定にした。作成したプログラムは、期待された様々な良い性質を満足し十分実用になることがわかった。また本論文では数値的安定化の方法を位相的歯止めをかけることと解釈したが、これにより他の幾何的アルゴリズムへの応用も考えやすくなったと思われ、この方面の研究に一つの方向を示すことができた。

謝辞

有益な御討論を頂いた東京大学工学部計数工学科伊理正夫教授、今井敏行助手に感謝します。本研究は文部省科学研究費補助金(一般研究(C), 課題番号01550279)の援助を受けている。

参考文献

- [1] Guibas, L. and Stolfi, J.: Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams. *ACM Transactions on Graphics*, Vol. 4 (1985), pp. 74-123.
- [2] 伊理(監), 腰塚(編), 他: 計算幾何学と地理情報処理. 共立出版, 東京, 1986.
- [3] 伊理, 杉原: 計算誤差を考慮した幾何的アルゴリズム. 情報処理学会アルゴリズム研究会研究報告, 88-AL-1-1, 1988.

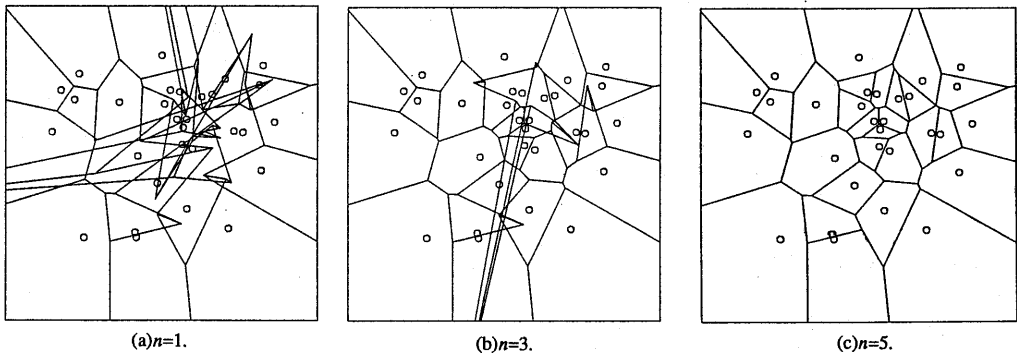


図 5.2 精度上昇による「収束」の様子.

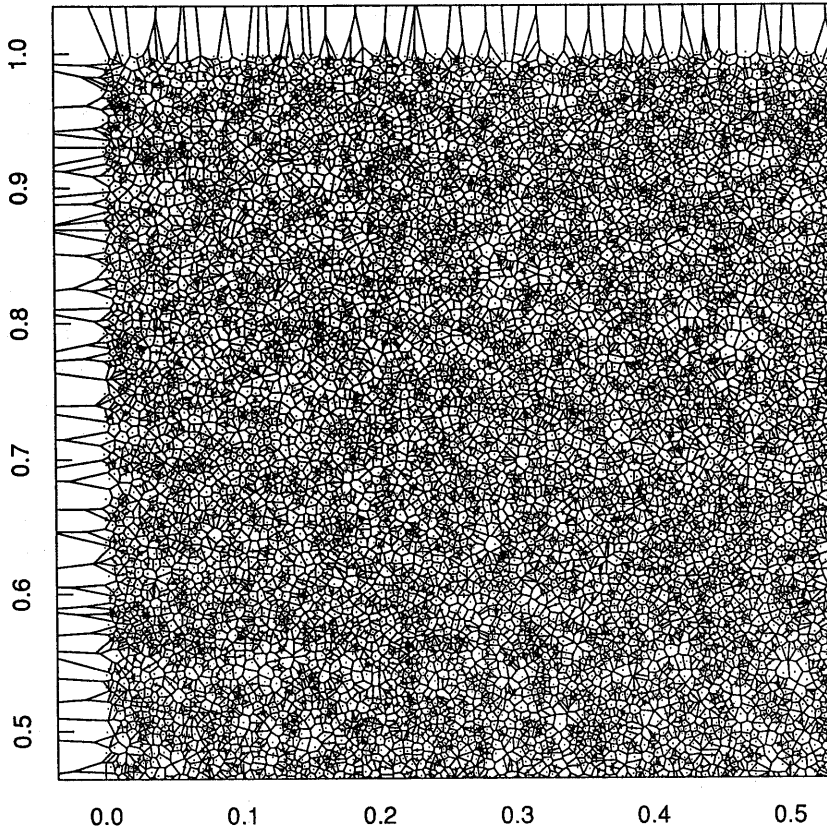


図 5.3 領域 $[0,1] \times [0,1]$ 中にランダムに分布する40000点について
単精度計算で求めたVoronoi図 (但しその左上の四分の一、
図中の黒点は母点を表す)。

[4] Katajainen, J. and Koppinen, M.: Constructing Delaunay Triangulations by Merging Buckets in Quadtree Order. Fundamenta Informaticae, Vol. 11 (1988), North-Holland, pp. 275-288.

[5] Lee, D. T. and Schachter, B. J.: Two Algorithms for Constructing a Delaunay Triangulation. International Journal of Computer and Information Sciences, Vol. 9 (1980), pp. 219-242.

[6] 大屋隆生: Voronoi線図の効率的構成法に関する研究。東京大学大学院工学系研究科情報工学専門課程修士論文, 1983.

[7] Ohya, T., Iri, M. and Murota, K.: Improvements of the Incremental Method for the Voronoi Diagram with Computational Comparison of Various Algorithms. Journal of the Operations Research Society of Japan, Vol. 27 (1984), pp. 306-336.

[8] 大石泰章: 分割統治型Voronoi図構成算法の数値的安定化。東京大学工学部計数工学科卒業論文, 1990.

[9] Sugihara, K. and Iri, M.: Construction of the Voronoi Diagram for One Million Generators in Single-Precision Arithmetic. Research Memorandum RMI 89-05, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo, 1988.