

## 計算誤差を考慮した幾何的アルゴリズム

伊理 正夫, 杉原 厚吉

東京大学工学部計数工学科

誤差を伴う計算のもとでもそれなりに正しく動作するアルゴリズムの概念を提唱し、ポロノイ図作成問題に対してその実例が構成できることを示す。実数データを扱う問題では、数値の表現誤差、計算の丸め誤差等のため、“理論的に正しい”アルゴリズムと実際に安定して動作するプログラムとの間には大きなギャップがある。このギャップを埋める試みの一つとして、計算を任意の精度で行っても真の解の位相的な属性の一部を満たす結果を必ず出力して停止し、精度が上がるにつれてこの出力が真の解に収束する、という性質をもつアルゴリズムを作ることには意義があるという考え方を提唱する。実例として、数値計算結果が位相的性質と矛盾したら位相的性質を優先させるという方針で、従来のポロノイ図作成アルゴリズムを上の意味でうまく動作するように修正できることを示し、そのプログラム例も紹介する。

## GEOMETRIC ALGORITHMS ROBUST AGAINST NUMERICAL ERRORS

Masao IRI 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

A new concept of "an algorithm that works well even if numerical errors arise in the course of processing" is proposed. An algorithm of this type always outputs a result that is consistent in a topological sense, and as the precision of arithmetic becomes higher, the result converges to the exact solution. As an example, an algorithm for constructing the Voronoi diagram is described together with its implementation, where inconsistencies caused by numerical errors are resolved by assigning the higher priority to the consistency of topological properties than to quantitative numerical values.

## 1. はじめに

今日、アルゴリズムについて論じるときには、まずその正しさを証明し、次にその計算量を評価するというのが常識となっている。確かに、問題の解が正しく得られることの保証と、その解を得るために要する計算時間および記憶場所の減少は、実用的なアルゴリズムを構成する際に考慮すべき重要なことである。実際、ある種のグラフ探索のように、アルゴリズムの正しさと計算量さえおさえられていれば実用上十分であるような問題も少なくない。しかし、一般に、アルゴリズムの正しさと計算量とだけに関する抽象化された世界に閉じこもっていたのでは、実用的な情報処理システムを作るという目的からは遊離してしまう危険性がある。

アルゴリズムを構成する本来の目的は、計算機で実際に動くプログラムを作り、それを利用することである。そのためには、数値計算誤差への対策、プログラム化の容易さなど、さまざまな側面を総合的に考慮しなければならない。たとえば、実数を扱う計算の途中で丸め誤差等が生じないという仮定のもとでアルゴリズムの正しさを証明しても、現実の計算機でそのアルゴリズムが正しく動くことの保証にはならない。また、技巧的なデータ構造を用いて計算複雑さのオーダを下げるのに成功しても、そのデータ構造のプログラム化が気の遠くなるほど複雑なものであったらやはり実用にはならない。にもかかわらず、アルゴリズムの理論では正しさと計算量という観点のみが強調される傾向がある。

計算誤差への配慮は、整数計算だけで解くことのできる組合せ問題などに対しては不要である場合もある。また、それとは対照的に、数値計算の分野では、計算誤差は中心的課題として取り上げられてきた。しかし、これら両者は、総合的なアルゴリズム構成という立場からみると、むしろ例外的な場合である。たとえば、計量と位相の両側面が存在する計算幾何学のような分野では、計算誤差の問題は、アルゴリズムの正しさという概念に関する根本的反省を我々に迫っているが、現状は、誤差を無視して作られたアルゴリズムに対してプログラム化の段階で小手先の工夫を加えているにすぎず、それでは根本的な誤差対策が図れるはずはなく、でき上がったプログラムは計算誤差による暴走・挫折の危険をはらんだまま使われているといつてよい。

以上の反省のもとに、本稿では、計算誤差の存在を前提としたアルゴリズム論の構成を試みる。すなわち、アルゴリズムの中で使われる個々の計算が誤差を伴うものであってもそれなりに正しく動作するアルゴリズムの概念を提唱し、そのようなアルゴリズムが実際に構成できることを、計算幾何学における代表的問題であるボロノイ図作成問題を例にとり示す。また、理論的には存在する退化等の多くの例外的ケースが誤差のある世界ではそもそも識別不可能であるということ

から、誤差を考慮したアルゴリズムは例外処理のない単純な構造をもちプログラム化が容易である、という一見逆説的な利点をもつことも併せて示す。

## 2. アルゴリズムの歴史と反省<sup>[1]</sup>

今世紀前半までのアルゴリズム論では、計算可能性（というよりむしろ計算不可能性）が主題であった。そこでは、ゲーデルの不完全性定理に代表されるように記号論理が万能ではないことを示す例がいくつも得られ、論理万能の楽観主義に警告を与えた。しかしこれらの成果は現実の工学的問題に直接かかわるものではなく、基本的にはアルゴリズム論は数学基礎論の世界に属するものであった<sup>[2]</sup>。

今世紀後半に入ると、計算量という物差でアルゴリズムの良さを計る考え方が定着し、アルゴリズムの理論にいくらかの工学的香りが加わってきた<sup>[3]</sup>。しかし、ひとたび物差が定着するとそれが一人歩きを始める。すなわち、その物差で計って計算量が小さいアルゴリズムが構成できれば、たとえそれが実用からはほど遠いものであっても“成果”であるとみなされる風潮が広まる。たとえば、2個の  $n \times n$  行列の積を  $O(n^3)$  より少ない計算量で求める一連のアルゴリズムの開発<sup>[4]</sup>などはそのような“成果”の代表例であろう。効率良くかつ正しく動くプログラムを実際に作ることを目指す工学的立場からは、アルゴリズム論は実際の計算の諸側面を考慮した総合的な理論でなければならない。

このような反省は、実際にいくつかの形で始まっている。その一つは、幾何学的アルゴリズムにおける退化回避法である。幾何学的対象を扱う際には、3点が1直線上に並んだり、4平面が1点を共有するなど、対象が特殊な位置関係に置かれたとき（このような状態は退化とよばれる）、さまざまな例外処理が必要とされる。そのような例外処理をすべて書き並べようとするアルゴリズムの構造が非常に複雑になってしまうため、退化が生じてもそれを退化していない状態に帰着させることによって例外処理の手間を省こうとする工夫が提案されつつある<sup>[5,6,7]</sup>。これらは、アルゴリズム論が計算量以外の側面を考慮し始めているという意味で歓迎すべき動きではある。しかし、そこでも、計算は誤差なく行われ、したがって退化が生じていることを正確に判定できるという仮定のもとで、議論がなされている点は残念である。

もう一つの反省の動きは、誤差に影響されない結果を得るために必要な計算精度を見積ってそれだけの高精度計算をしようという考え方である。アルゴリズムで処理すべき現実のデータは有限の精度で与えられるから、可能なデータは有限個しかない。したがって、それらに演算を施した結果も有限・離散的であり、それらを互いに区別するためにはある有限の精度で計算すれば十分なはずである。この考え方に基づいて実際に必要な計算精度を見積るための一般論の試み<sup>[8,9]</sup>や、それをソリッドモデリングなどの具体的問題に応用す

る試み<sup>[10,11]</sup>などがなされている。しかし、この方法も計算精度を低くおさえることのできる場面では有効であるが、必要な計算精度が現実離れしたほど大きくなってしまふ場面では役に立たない。

では、普通の精度（たとえば単精度浮動小数点）で計算を行っても破綻せず、しかもプログラム化が容易で、計算効率も良いアルゴリズムというものは、存在しえないであろうか。このような望みが夢ではないことを以下で示したい。

### 3. ボロノイ図と計算誤差

2次元平面  $R^2$  上の互いに異なる  $n$  個の点  $p_1, \dots, p_n$  に対して、領域

$$V(p_i) = \{p \mid p \in R^2, d(p, p_i) \leq d(p, p_j), j = 1, \dots, n, j \neq i\}$$

を  $p_i$  のボロノイ領域とよぶ。ただし  $d(p, q)$  は点  $p$  と点  $q$  のユークリッド距離を表す。 $\{V(p_1), \dots, V(p_n)\}$  は平面の一つの分割をなし、この分割のことを  $p_1, \dots, p_n$  を母点とするボロノイ図という。2個のボロノイ領域の共通の境界をボロノイ辺、3個以上のボロノイ領域が共有する点をボロノイ点とよぶ。ボロノイ辺は2個の母点の垂直二等分線であり、ボロノイ点は3個の母点を通る円の中心である。ボロノイ図は計算幾何学における基本的概念の一つで<sup>[12]</sup>、地理的最適化<sup>[13]</sup>、パターン認識<sup>[14]</sup>など多方面の応用もっている。

ボロノイ図を構成する実用的算法として最も成功しているのは、逐次添加法<sup>[15]</sup>であろう。これは、2個または3個の母点に対するボロノイ図から出発して、母点を1個ずつ追加しながら、すでに得られているボロノイ図を更新していくという方法である。図1(a)は、黒丸の母点に対する実線のボロノイ図に、白丸で示された新たな母点  $p_i$  が追加されたときの更新の様子を示したものである。まず  $p_i$  に最も近い母点（この場合は  $p_j$ ）を求め、 $p_i$  と  $p_j$  の垂直二等分線（これが新しくボロノイ辺となる）を  $V(p_j)$  の境界と交わる点  $q$  まで延ばす。次に、 $q$  で  $V(p_j)$  に隣接するボロノイ領域の母点  $p_k$  と  $p_i$  の間の垂直二等分線を作り、 $V(p_k)$  の境界まで延ばす。再び領域  $V(p_j)$  に戻るまで同様のことを繰り返す。今作ってきた垂直二等分線列（図1(a)の破線で示した線列）の内部を母点  $p_i$  の領域  $V(p_i)$  とすればよい。この方法は、ボロノイ図を構成する他のアルゴリズム

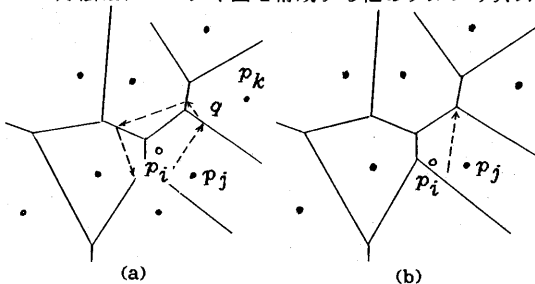


図1. 逐次添加法と計算誤差

<sup>[12,16]</sup>と比べて実際的には効率が良く数値的にも安定であり、母点の添加順序を工夫することによって母点1個当たり平均  $O(1)$  の処理速度を達成するプログラムが、この方法に基づいて実際に作られ、使われている。

しかし、この方法も、数値誤差に対して弱い面をもっている。たとえば図1(a)で新しく追加された母点  $p_i$  の位置が少し変わると、(b)のように、 $p_i$  と  $p_j$  の垂直二等分線がボロノイ点の近くを通過する場合がある。このとき、このボロノイ点のどちら側を通過するかの判定は計算誤差に依存し、それが他の判定結果と相対的に矛盾することがある。たとえば、このボロノイ点の垂直二等分線より  $p_i$  側にあると判定されると、次に  $p_i$  と  $p_k$  の垂直二等分線を調べることになるが、それが  $V(p_k)$  の境界と交差しているか否かの判定も計算誤差に依存するため、交差していないという矛盾した判定結果が得られるかもしれない。

もちろん、プログラムを作るときには、このような判定誤りが生じても簡単には転ばないよう多くの対策が講じられる。しかし、*ad hoc* な対策をいくら積み重ねても判定誤りのすべての場合に対処し得るという保証を与えることは難しい。

## 4. 誤差のある世界で望み得ること

### 4.1. ボロノイ図の表現法

ボロノイ図は、辺が途中で交差することなく平面に描かれたグラフとみなすことができる。このグラフから計量的属性を取り除いた構造をボロノイ図の接続構造とよび、 $G$  で表す。すなわち、 $G$  は平面に位相的に埋め込まれた平面グラフであり、それぞれの辺が平面のどの位置に描かれているかは問題にしない。 $G$  も平面を  $n$  個の部分に分割する。このように分割された部分をセルとよび、対応する母点の番号で名付けることにする。セルは、ボロノイ領域と異なり、位置や形状という属性はもたず、他のセルとの接続関係だけが意味をもっている。以下の議論の便宜のために、ここではボロノイ図を表現する際に次の二つの便法を採用する。

第一に、母点の存在する領域を取り囲む十分遠方の閉曲線を導入し、無限に延びたボロノイ辺は仮想的にこの閉曲線に交差しているものとみなす。この閉曲線もボロノイ辺に加えることによって、接続構造  $G$  が表すグラフのすべての頂点の次数が3以上となる。図2(a)のボロノイ図の接続構造をこの便法を用いて表したのが(b)である。この便法によって、平面は  $n+1$  個の領域に分割されることになるが、外周閉曲線の外側のセルには番号0を割り当てることとする。

第二の便法は、次数が4以上のボロノイ点が見れたら、それを次数3の複数の点に分解することである。ボロノイ点は3個の母点を頂点とする三角形の外心であるから、4個以上の母点が同一円周上にあるときに  $G$  において次数が4以上の頂点が見れることがある。図2(a)の母点  $p_4, p_5, p_6, p_7$  のボロノイ領域が

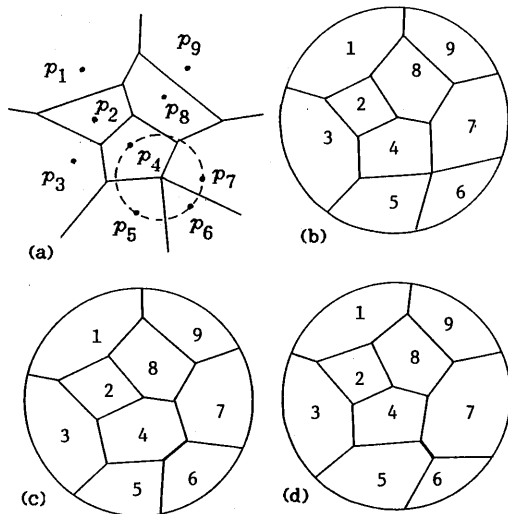


図2. ボロノイ図の接続構造を次数3の平面グラフで表すための便法: (a) ボロノイ図, (b) 無限遠方閉曲線の導入, (c), (d) 次数4の頂点の解消.

共有する点がある。このような点は、長さ0のボロノイ辺を必要な数だけ導入することによって、次数3の2個以上の点に分解するものとする。これは、直観的には、母点位置に無限小の摂動を加えて4個以上の母点が同一円周上に載っているという性質を崩してからボロノイ図を作りなおすことに対応している。このときの摂動の加え方には任意性があり、その結果得られる接続構造は一義的には決まらない。図2(a)のボロノイ図に対しては(c)と(d)の二通りの表現が可能である。

以上の二つの便法により、接続構造の任意の頂点はちょうど3個のセルで囲まれることになる。これらのセルの番号が反時計回りに  $i, j, k$  の順であるとき、この頂点を  $q_{ijk}$  と書くことにする。計量的な性質を問題にするときには、 $q_{ijk}$  を3個の母点  $p_i, p_j, p_k$  のボロノイ領域がこの順に反時計回りに囲むボロノイ点と同一視する。

#### 4. 2. 有限精度計算のもとでのボロノイ図作成アルゴリズム

ボロノイ図が満たすべき性質を思い付くままに並べると、次のようなものがある。

- (1)\* 接続構造  $G$  は平面に相対的に埋め込まれた次数3の平面グラフである。
- (2)  $G$  は外周閉曲線の内部を  $n$  個のセルに分割し、母点とセルの間には1対1対応がある。
- (3)\* 二つのボロノイ領域が共有するボロノイ辺は高々1個である。

\* 実際には、ボロノイ図はグラフとしては次数3の3連結平面グラフであることが知られている。(3)は、このことから導かれる性質である。

(4) ボロノイ点はそれにかかわる3個の母点を通る円の中心である。

(5) ボロノイ辺はそれにかかわる2個の母点の垂直二等分線の一部である。

(6) 各ボロノイ領域は凸多角形である。

(7) 各母点是对应するボロノイ領域の内部にある。

(8) ボロノイ辺は互いに途中で交差することはない。

このうち (1), (2), (3) は接続構造  $G$  に関する位相的性質であり、(4)と(5)は  $G$  が与えられたときボロノイ図を描くための方法を与えているとみなすことができる。一方、(6), (7), (8) はそのようにして描いたボロノイ図が満たすべき性質である。

以下では、ボロノイ図の作成とは「母点を与えられたとき接続構造  $G$  を決定すること」であるとみなす。なぜなら、いったん正しい  $G$  が得られれば、(4), (5)の性質を利用して実際に図を描くことができるからである。

ところで、誤差を伴う計算によってボロノイ図を作ろうとすると、一般に、計量的な性質を厳密に満足させることはできない。すなわち、(6), (7), (8)を厳密に満たすアルゴリズムを望むことは不可能であると認める立場をとろう。すると、任意の精度の計算で望み得るのは計量を全く伴わない(1), (2), (3)の性質のみである。そこで、計算誤差を伴う世界におけるボロノイ図作成アルゴリズムとは、「その中で使われる実数計算の精度がどのようなものであると、常に(1), (2), (3)を満たす何らかの接続構造を出力し、計算精度が高くなるに従って出力が真のボロノイ図の接続構造に“収束”していく」という性質をもったものであるとするのが自然であろう。そこで、我々は次のような定義を提唱する。

ボロノイ図作成アルゴリズムへの入力は、母点数  $n$  および母点の  $x, y$  座標  $x_1, y_1, \dots, x_n, y_n$  である。これらをまとめて  $d = (n, x_1, y_1, \dots, x_n, y_n)$  とおく。真のボロノイ図に対応する接続構造(次数4以上のボロノイ頂点を解消する際の任意性があつたから、これは一意とはかぎらない)の集合を  $A(d)$  とし、性質(1), (2), (3)を満たす接続構造の集合を  $\bar{A}(d)$  とする。“ボロノイ図構成アルゴリズム”であることが望まれている一つのアルゴリズムの中で使われる任意の実数値関数  $f(x)$  に対して「 $f(x)$  の計算値を  $\tilde{f}(x)$  と書くとき、 $\tilde{f}(x) \geq e$  あるいは  $\tilde{f}(x) \leq -e$  なら  $f(x)$  の符号は正しく判定され、 $|\tilde{f}(x)| < e$  のときには  $f(x)$  の符号は正しく判定されることも誤って判定されることもどちらもあり得る」という性質をもった正数  $e$  が存在すると仮定する。 $e$  はアルゴリズム中で実行される実数計算の精度を示すパラメータで、 $e$  が大きければ精度が低く、小さければ精度が高い。(  $e$  はマシン・エプシロンのようなものではなく、アルゴリズムにも依る。 ) このアルゴリズムに入力  $d$  が与えられたとき、上記のような符号判定の曖昧さのもとであらゆる可能な出力の集合を  $A(d, e)$  と書く。

ポロノイ図作成アルゴリズムは次の性質(9), (10)を満たすとき計算誤差に対してロバスト(頑健)であるという。

(9) 任意の入力  $d$ , 任意の計算精度  $e$  に対して  $A(d, e) \subseteq \bar{A}(d)$  が成り立ち, アルゴリズムは必ず  $A(d, e)$  の一つの要素を出力して停止する。

(10)  $\lim_{e \rightarrow 0} A(d, e) \subseteq A(d)$ .

(9)はアルゴリズムが性質(1), (2), (3)をみたく接続構造を出力して必ず停止することを要求し, (10)は計算精度を上げると出力が真のポロノイ図に収束することを要求している。以下では, このような意味でロバストなアルゴリズムが構成できることを示す。

## 5. ロバストなアルゴリズム

### 5.1. 位相構造の優先

処理の途中で計量的な判定と位相構造とが矛盾するときには, 位相構造の一貫性を優先させることにする。これを実現するために, ここでは次の性質に着目する。すなわち, 母点逐次添加法において, 新しい母点の一つが追加されたとき取り除かれるべきポロノイ頂点およびそれと接続するポロノイ辺で定められる部分グラフを  $T$  とすると,

(11)  $T$  は木(すなわち連結な無閉路グラフ)であり,

(12) 任意のポロノイ領域に対して, その境界が  $T$  と共通部分をもてばその共通部分は連結している。

なぜならば, 部分グラフ  $T$  が非連結であると新しい母点に対するポロノイ領域が2個以上の連結成分から構成されることになり, 閉路を含むと今まで存在していたポロノイ領域の一つが完全に消えてしまうことになり, あるポロノイ領域の境界と  $T$  の共通部分が非連結であると, そのポロノイ領域と新しく追加された母点のポロノイ領域が2個以上のポロノイ辺を共有してしまうことになり, いずれもポロノイ図の性質に反するからである。(十分遠方の閉曲線を導入して接続構造を閉じたグラフにしたため, 性質(11), (12)が無限ポロノイ辺に対しても成り立っていることに注意されたい。)一方, 性質(1), (2), (3)を満たす接続構造に対して取り除かれる部分グラフが(11), (12)を満たせば, 更新の結果得られる接続構造もやはり性質(1), (2), (3)を満たす。したがって, 除くべき部分グラフに性質(11), (12)さえ満足させておけば, 常に(1), (2), (3)を満たす接続構造が構成できる。以下では, この方針に基づいてロバストなアルゴリズムを作る。

### 5.2. 取り除くべき頂点の識別

母点  $p_1, \dots, p_{s-1}$  に関するポロノイ図  $V_{s-1}$  がすでに得られ, 母点  $p_s$  を新たに添加して  $p_1, \dots, p_s$  に関するポロノイ図  $V_s$  を作る場面を考えよう。 $V_{s-1}$  のポロノイ点  $q_{ijk}$  が  $p_s$  のポロノイ領域に入る(したがって取り除かれるべき点である)のは, 3個の母点  $p_i, p_j, p_k$  を通る円の内部に  $p_s$  が含まれるときである(そしてそのときに限る)から,

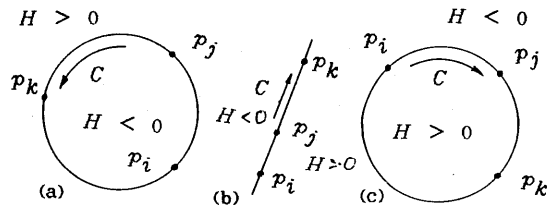


図3. 有向円周  $C$  に対する点  $p$  の位置関係と  $H(p_i, p_j, p_k, p)$  (図中では  $H$  と略記してある)の符号

これが判定できればよい。

点  $p$  の座標を  $(x, y)$  とし

$$(13) \quad h(x, y) = \begin{vmatrix} H(p_i, p_j, p_k, p) \\ 1 & x_i & y_i & (x_i^2 + y_i^2)/2 \\ 1 & x_j & y_j & (x_j^2 + y_j^2)/2 \\ 1 & x_k & y_k & (x_k^2 + y_k^2)/2 \\ 1 & x & y & (x^2 + y^2)/2 \end{vmatrix}$$

とおく。この多項式  $h(x, y)$  は  $x, y$  に関して2次で, 2次の項は  $c(x^2 + y^2)$  の形をしており,  $p = p_i, p_j, p_k$  のとき  $h(x, y) = 0$  となるから,  $h(x, y)$  の零点全体は3点  $p_i, p_j, p_k$  を通る円を表す。

( $p_i, p_j, p_k$  が同一直線上に並ぶときには直線を表すが, これも無限遠方の両端がつながった“円”とみなす。)この円の周に3点  $p_i, p_j, p_k$  をこの順に通過するような向きを与えたものを  $C$  と書くと,  $H(p_i, p_j, p_k, p) > 0, < 0$  あるいは  $= 0$  に従って,  $p$  はそれぞれ有向円周  $C$  の右側, 左側, あるいは上にある(図3参照)。今,  $p_i, p_j, p_k$  はこの順に反時計回りに頂点  $q_{ijk}$  を囲んでいるとしたから,  $q_{ijk}$  は  $H(p_i, p_j, p_k, p_s) < 0$  のとき取り除かれるべき頂点の一つであると判定することができる。

以上は  $q_{ijk}$  が有限のポロノイ点の場合であるが, 接続構造の外周閉曲線上の頂点(無限に延びたポロノイ辺の無限遠方の端)に対しても同じ形式の判定が可能である。 $w_1 = w_2 = \dots = w_n = 1$  とし, 母点  $p_j$  の位置を同次座標  $(x_j, y_j, w_j)$  で表す。また  $x_0^2 + y_0^2 \neq 0, w_0 = 0$  となる点  $p_0 = (x_0, y_0, w_0)$  を形式的に導入し, 接続構造の外周の外側のセルをこの  $p_0$  に対応させる。そして, (13)式のかわりに

$$(14) \quad \bar{H}(p_i, p_j, p_k, p) = \begin{vmatrix} w_i^2 & w_i x_i & w_i y_i & (x_i^2 + y_i^2)/2 \\ w_j^2 & w_j x_j & w_j y_j & (x_j^2 + y_j^2)/2 \\ w_k^2 & w_k x_k & w_k y_k & (x_k^2 + y_k^2)/2 \\ w^2 & wx & wy & (x^2 + y^2)/2 \end{vmatrix}$$

を考える。 $i, j, k$  がいずれも0でないときには(14)式は(13)式に帰着される。一方,  $i, j, k$  のうちの 하나가0のときは, (14)式の零点全体は残りの二つの母点を通る直線を表し, 点  $p$  がこの直線のどち

ら側にあるかが(14)式の符号から判定できる。そして、これが除くべき頂点の判定にもなっている。たとえば  $k = 0$  のとき、(14)式が正ならば  $p$  は  $p_i$  と  $p_j$  をこの順に通過する有向直線  $L$  の右側にあり、したがって点  $p$  の位置に新しい母点を添加したとき頂点  $q_{ij0}$  は残るのに対して、(14)式が負であれば  $p$  は  $L$  の左側となり頂点  $q_{ij0}$  は除かれるべき頂点の一つとなる。このように  $\bar{H}$  の符号を用いると、有限の位置を占めるポロノイ点と無限に延びたポロノイ辺の端とを区別しないで処理でき、例外処理の煩わしさを除くことができる。

ところで、 $\bar{H}(p_i, p_j, p_k, p_s) = 0$  のときもここでは特別扱いはしない。すなわちこの値が負なら  $q_{ijk}$  は取り除かれるべき頂点であるとみなし、正または零のとき残される頂点であるとみなす。誤差が入る計算のもとでは、 $\bar{H}$  の値がたまたま 0 になっても、それには  $|\bar{H}|$  が非常に小さくなったということ以上に特に意味がなく、性質(11)、(12)を優先する限りこのような判定で十分である。

### 5.3. アルゴリズム

最初の2個の母点  $p_1, p_2$  に関するポロノイ図は図4に示す接続構造をもつ。そこでこれから出発してポロノイ図を更新していく。アルゴリズムの基本的手順は次のとおりである。

1. 図4の接続構造を  $G$  とおく。

2.  $s = 3, 4, \dots, n$  の順に 2.1~2.4 を実行する。

2.1.  $p_1, \dots, p_{s-1}$  のうち  $p_s$  に最も近い母点  $p_i$  をみつける。

2.2.  $p_i$  を囲むポロノイ点のうち  $\bar{H}(p_i, p_j, p_k, p_s)$  の値が最も小さいもの  $q_{ijk}$  をみつけ、 $T = \{q_{ijk}\}$  とおく。

2.3.  $T$  に接続した各ポロノイ点  $q_{\alpha\beta\gamma}$  に対して  $\bar{H}(p_\alpha, p_\beta, p_\gamma, p_s) < 0$  であり、かつこの頂点の追加によって性質(11)、(12)が乱されないなら、この頂点を  $T$  に加える。これを、 $T$  がこれ以上大きくできなくなるまで繰り返す。

2.4.  $G$  において、 $T$  に属す頂点と  $T$  に属さない頂点を結ぶ辺を反時計回りに並べたものを  $e_1, \dots, e_m$  とする。これらの辺を途中で新しい頂点  $v_1, \dots, v_m$  を作って2分割し、 $T$  に属す頂点とそれに接続している辺を除き、辺  $(v_1, v_2), \dots, (v_{m-1}, v_m), (v_m, v_1)$  を加えて、これらで囲まれる新しいセルを  $s$  とする。

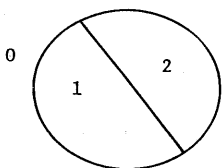


図4. 2個の母点  $p_i, p_j$  に対するポロノイ図の接続構造

ステップ1は初期化で、ステップ2が母点の添加である。2.1, 2.2は  $p_s$  の添加のための初期化で、母点  $p_s$  の添加によって除かれるべき頂点の一つをみつける。2.3で除くべき部分構造を構成し、2.4でそれを取り除いて新しいセルに置き換える。

このアルゴリズムでは、性質(11)、(12)を満たす部分グラフ  $T$  を、 $T$  と  $T$  に属さない頂点をつなぐ辺の途中で切断することによって除き、一つの閉路で置き換えているから、その出力は性質(11)、(12)を満たす。極端に言えば、2.2で求める最初の頂点をランダムに選び、2.3の  $\bar{H}$  の符号を乱数で決めたとしても、途中で行き詰まることなく性質(1)、(2)、(3)を満たす接続構造を作って停止する。また計算が十分高い精度で実行されれば、4個の母点在同一円周上に載ったときにはあとから添加された母点をこの円の外側へ微量だけずらすという摂動を加えたときに相当する接続構造が得られる。詳しい議論は省くが、このような考察からこれがロバストなアルゴリズムになっていることが証明できる[17]。

計算量の観点からみたとき、このアルゴリズムは従来のものとほとんど変わらない速度をもっている。1, 2.2は定数時間内に実行でき、2.4は取り除かれる部分構造の大きさに比例する手間で実行できる。また、いくらかの自明でない工夫を施すことにより、2.3の手続きも取り除かれる部分構造の大きさに比例する手間で実行できる[17]。したがって、文献[15]でなされている添加順序に関する工夫を採用すれば、そこでのアルゴリズムと同様、本アルゴリズムも母点1点当たり平均  $O(1)$  の処理時間が達成できる。

### 6. プログラムの振舞い

本アルゴリズムを PC-FORTRAN で書き、NEC PC-9801VX で走らせてその振舞いをみた。1988年4月末現在の版は、21個の副プログラムから成り、コメント文、デバッグ用の処理文などをすべて含めて約1200行の大きさである。これにテスト用の母点発生機能、ポロノイ図表示機能などを加えて約2000行のプログラムを作り、母点数2000の計算が十分可能であるよう記憶場所を確保したところ、実行可能プログラムの大きさは約420Kバイトであった。ただし浮動小数点計算はすべて単精度である。このプログラムの振舞いの例を以下に示す。

(i) ランダムに配置された母点に対する計算誤差の影響

図5(a)は、ランダムに発生した20個の母点に対する本プログラムの出力で正しいポロノイ図となっている。(b)は  $\bar{H}$  の計算にある程度の誤差を加えた場合の出力である。ここでは、単精度浮動小数点で求めた  $\bar{H}$  の各値に区間  $[-r, r]$  の一様乱数の値を加え合わせてから符号を判定した。(c)は一様乱数の発生区間をさらに大きくとった場合の出力である。(d)は  $\bar{H}$  の符号を乱数で決めたとときの出力である。いずれの場合もプログラムは途中で行き詰まることなく結果を出力し

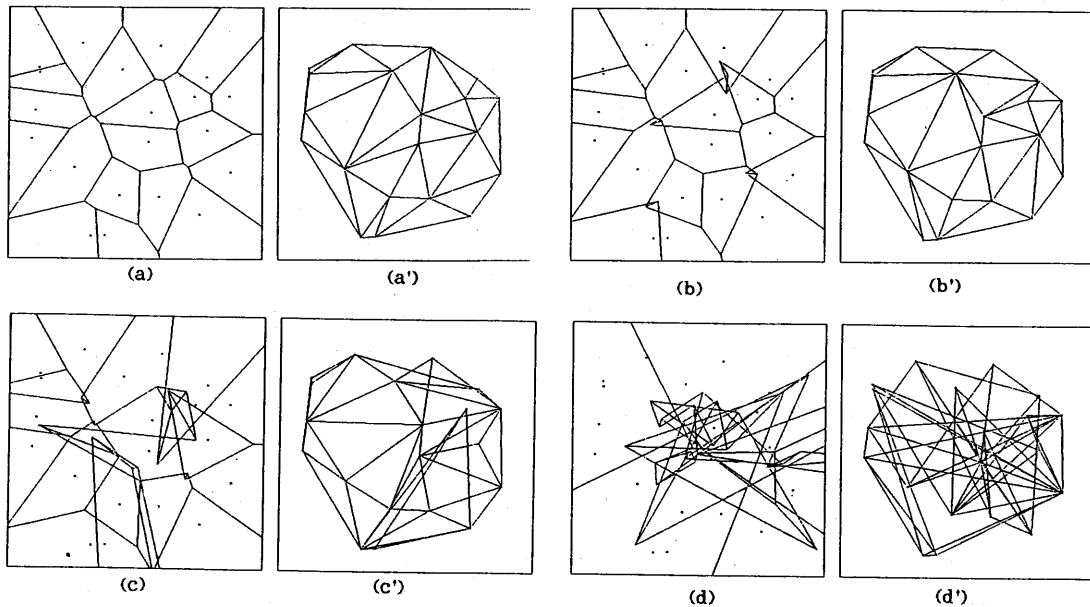


図5. 単位正方形の中にランダムに配置された母点に対するポロノイ図と計算誤差を入れた場合の出力  
 (a),(a'):  $r=0$ ; (b),(b'):  $r=2^{-8}$ ; (c),(c'):  $r=2^{-7}$ ; (d),(d'):  $\bar{H}$  の符号を乱数で決める

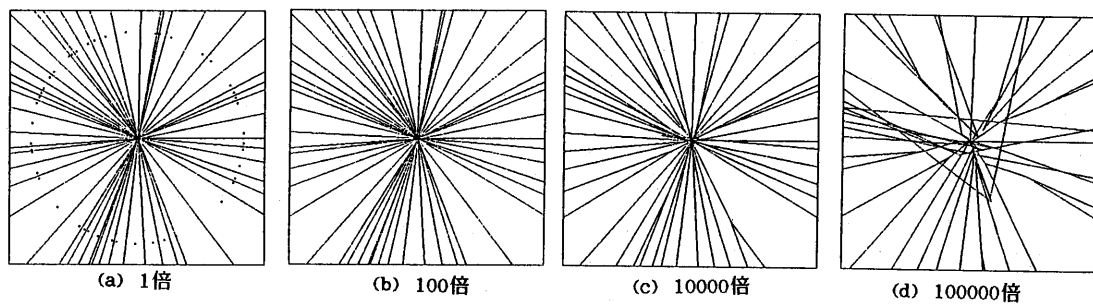


図6. 円周上にランダムに発生させた50個の母点に対する出力と、その中央付近の拡大図

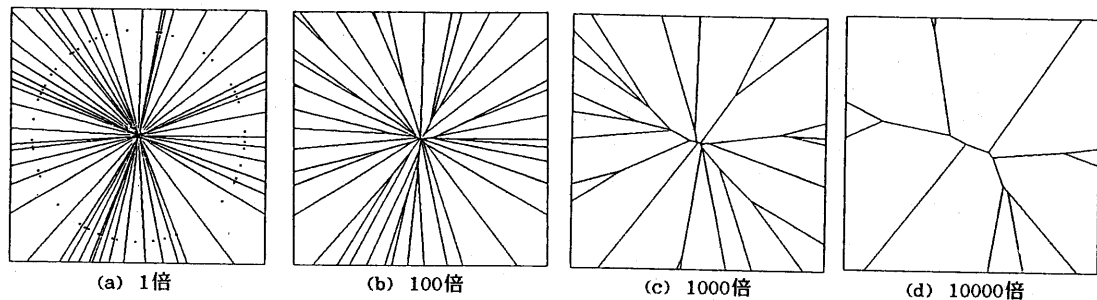


図7. 円周上から半径方向へわずかに摂動を加えた50個の母点に対する出力と、その中央付近の拡大図

た。これらすべての接続構造が次数3の平面グラフであることは手作業で確認済みである。(a'), (b'), (c'), (d')は、それぞれ (a), (b), (c), (d)の双対グラフ(すなわち、母点を頂点とし、セルの境界を共有する母点同士を辺で結んでできるグラフ; 正しいポロノイ図に対応するもの(a')はドロネ網とよばれる)である。

(ii) 同一円周上の母点に対する振舞い

図6(a)は、同一円周上にランダムに発生させた50個の母点に対する出力結果である。(b), (c), (d)はそれぞれ(a)の中央付近の百倍、一万倍、十万倍の拡大図である。(a)は一見すると正しいポロノイ図のようにみえるが、(c)や(d)からわかるように、中央付近は真のポロノイ図からはかなりかけ離れている。この母点の配置はポロノイ図作成にとって非常に意地悪なデータであり、この程度の乱れが単精度計算で生じるのは当然であろう。むしろ、このようなデータに対しても最後まで処理が進んで結果が出力されていることに注目すべきである。

図7は、図6の実験で用いた母点をそれが載っている円の半径方向に半径の1万分の1の幅の一樣乱数を用いて移動させたデータに対する出力である。(b), (c), (d)は、それぞれ(a)の中央付近の百倍、千倍、一万倍の拡大図である。わずかではあるが同一円周上からはずしたことによって意地悪なデータではなくなり、正しくポロノイ図が計算できていることがわかる。

## 7. おわりに

計算誤差を伴う世界でそれなりに動作するアルゴリズムの概念を提唱し、ポロノイ図作成問題に対してそのようなアルゴリズムの存在することを示した。今後は、同様のアルゴリズムを計算幾何学の種々の問題に対して構成していきながら、計算誤差対策の煩わしさからプログラムを解放するための方法論を確立したい。

本研究は、セコム科学技術振興財団、文部省科学研究費補助金(課題番号 62580017)の援助を受けている。

## 参考文献

- [1] 伊理正夫: 最近のアルゴリズム特集1, 総論. 電子通信学会誌, vol. 69 (1986), pp. 290-299.
- [2] M. Davis: *Computability and Unsolvability*. McGraw-Hill, New York, 1958.
- [3] A. V. Aho, J. E. Hopcroft and J. D. Ullman: *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, 1974.
- [4] V. Pan: *How to Multiply Matrices Faster*. Lecture Notes in Computer Science, 179, Springer-Verlag, Berlin, 1984.
- [5] H. Edelsbrunner: Edge-skeletons in arrangements with applications. *Algorithmica*, vol. 1 (1986), pp. 93-109.

- [6] H. Edelsbrunner and E. P. Mücke: Simulation of simplicity — A technique to cope with degenerate cases in geometric algorithms. Report no. UIUCDCS-R-88-1419, University of Illinois at Urbana-Champaign, 1988.
- [7] C.-K. Yap: Symbolic treatment of geometric degeneracies. Preprint, New York University, 1987.
- [8] 吉田清範: 代数的な量の符号判定に必要な計算精度. 電子通信学会論文誌, vol. J69-A (1986), pp. 543-547.
- [9] 吉田清範: 初等的図形の交差判定法 — 計算量および計算精度の立場から. 東京大学大学院工学系研究科計数工学専攻学位論文, 1988.
- [10] 杉原厚吉, 伊理正夫: 計算誤差による暴走の心配のないソリッドモデラの提案. 情報処理学会論文誌, vol. 28 (1987), pp. 962-974.
- [11] T. Ottman, G. Thiemt and C. Ullrich: Numerical stability of geometric algorithms. *Proceedings of the 3rd ACM Conference on Computational Geometry*, Waterloo, 1987, pp. 119-125.
- [12] F. P. Preparata and M. Shamos: *Computational Geometry — An Introduction*. Springer-Verlag, New York, 1985.
- [13] 伊理正夫(監), 腰塚武志(編), 他: 計算幾何学と地理情報処理. 共立出版, 東京, 1986.
- [14] G. T. Toussaint: Pattern recognition and geometric complexity. *Proceedings of the 5th International Conference on Pattern Recognition*, 1980, pp. 1324-1347.
- [15] T. Ohya, M. Iri and K. Murota: 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.
- [16] S. Fortune: A sweeping algorithm for Voronoi diagrams. *Proceedings of the 2nd ACM Annual Symposium on Computational Geometry*, Yorktown Heights, 1986, pp. 313-322.
- [17] 伊理正夫, 杉原厚吉: 計算誤差があっても破綻しないポロノイ図作成アルゴリズム. 準備中.