

真円度問題

—計算幾何学のアプローチ—

榎原 博之

1. はじめに

機械部品の精度として重要な幾何公差（形状公差）の1つに真円度[15]と呼ばれるものがある。これは、「円筒断面の輪郭を2つの同心円で挟んだとき、両円の間隔が最小となる両円の半径差」と定義されている[8]。真円度は、円形状の機械部品—たとえば、ボールベアリング、エンジンのシリンダ、ビデオテープレコーダのロータリーヘッドなど—の精度を測定するためにはなくてはならないものであり、真円度測定機と呼ばれる装置によって測られる。

真円度測定機では、一般に輪郭形状を有限個の点集合に置き換えて計算機で処理するため、筆者らはこの真円度を求める問題を真円度問題として次のように定式化した[2]。

真円度問題：平面上に n 個の点が与えられたとき、同心の2つの円を考え、その2つの円の間すべての点が含まれるという条件のもとで、その2つの円の半径差を最小にする問題(図1)

この問題はある種の設備配置問題と考えることができ応用の多い問題であると考えられる。

この問題に対する古くから使われている解法は、最小2乗法による解法、Min-Max法を適用した解法、シンプレックス法を用いた解法等があるが、近似解法に過ぎないとか、厳密解法であっても時間計算量に難点があるものばかりであった[10][14][15]。

そこで、筆者らは最近アルゴリズム理論の分野で非常に注目されている計算幾何学[1][7][11](特に、ポロノイ図[7][11][12])をこの問題に適用することを考えた[3]。計算幾何学とは、幾何学に計算の複雑さの理論を導入し、幾何学的な問題を計算機で効率よく処理するアルゴリズムを開発するとともに、その限界を究明する研究分野である。

えばら ひろゆき 大阪大学 工学部 通信工学科
〒565 吹田市山田丘 2-1

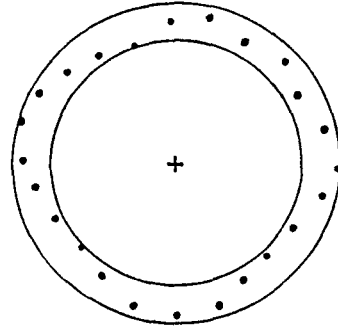


図1 真円度問題

本稿では、筆者らが提案した真円度問題に対する多項式時間のアルゴリズムを述べ、それを実用的に効率の良いものにするために、前処理として解の厳密性を保証したまま不要点を削除するアルゴリズムを提案し、その実用性を計算機実験を通して示す。

2. 真円度アルゴリズム

2.1 ポロノイ図(勢力圏図)

ポロノイ図とは、計算幾何学において非常に有名な概念で、近さを効率良く表わす一種のデータ構造と考えることができる。真円度問題だけでなく、各種の施設配置問題に利用することが可能であり、現にいくつかの問題に利用されている[12]。

われわれの扱うポロノイ図は、ユークリッド距離 (L_2 距離) における最近点のポロノイ図(一般のポロノイ図)と最遠点のポロノイ図であるが、 k 番目までに近い点のポロノイ図や、マンハッタン距離 (L_1 距離) や L_∞ 距離などに対するものもある。また、点ばかりでなく直線や多角形などに対するものもある。

2.1.1 最近点のポロノイ図

最近点のポロノイ図(勢力圏図)は、次のように定義されるものである[12]。

「平面上の n 個の点 $p_i = (x_i, y_i)$ ($i=1, 2, \dots, n$) に対

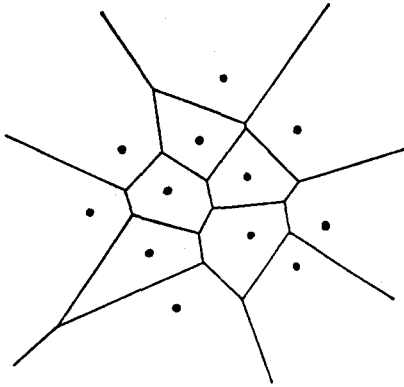


図 2 最近点のボロノイ図

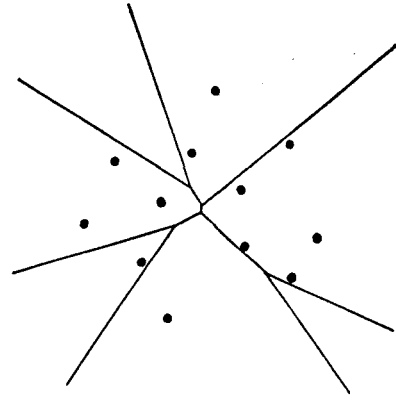


図 3 最遠点のボロノイ図

して、点 p_i の '勢力圏' $V(p_i)$ を

$$V(p_i) = \bigcap_{j=1}^n \{p=(x, y) \mid d(p_i, p) \leq d(p_j, p)\}$$

と定め、 $V(p_i)$ ($i=1, 2, \dots, n$) による平面分割を最近点のボロノイ図と呼ぶ (図 2)。」ここで、 $d(p_i, p)$ は 2 点 p_i, p のユークリッド距離、すなわち、 $\sqrt{(x-x_i)^2 + (y-y_i)^2}$ である。

ボロノイ図において、勢力圏 $V(p_i)$ を点 p_i のボロノイ領域と呼び、これは凸多角形である。また、ボロノイ領域の頂点をボロノイ点、辺をボロノイ辺と呼ぶ。

2.1.2 最遠点のボロノイ図

最遠点のボロノイ図(非勢力圏図)は、次のように定義されるものである[12]。

「平面上の n 個の点 $p_i=(x_i, y_i)$ ($i=1, 2, \dots, n$) に対して、点 p_i の '非勢力圏' $U(p_i)$ を

$$U(p_i) = \bigcap_{j=1}^n \{p=(x, y) \mid d(p_i, p) \geq d(p_j, p)\}$$

と定め、 $U(p_i)$ ($i=1, 2, \dots, n$) による平面分割を最遠点のボロノイ図と呼ぶ (図 3)。」最遠点のボロノイ図の場合はすべての点が領域を持つわけではなく、与えられた点集合の凸包上の点についてのみ領域を持つ。

これら 2 つのボロノイ図を構成するアルゴリズムは、文献[12]で提案されている。その時間計算量は、分割統治法を使うことにより $O(n \log n)$ である。ただし、われわれの計算機実験の際には、平均的には計算時間が短い文献[6]の逐次添加法を用いた。この時間計算量は $O(n^2)$ である。

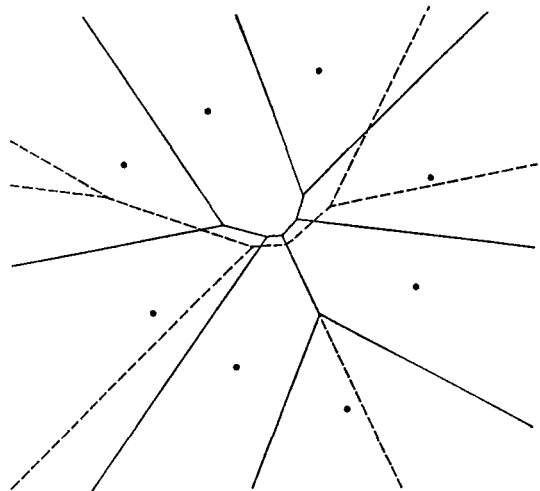
2.2 真円度アルゴリズム

真円度問題において、真円度を決定する同心円の中心はその中心から最も近い点 (1 点とは限らない) と最も遠

い点 (1 点とは限らない) の距離の差が最小になる点である。

ところで、与えられた点についてのボロノイ図を考えると、最近点のボロノイ領域内の任意の点はその領域に対応する点が最近点であり、最遠点のボロノイ領域内の任意の点はその領域に対応する点が最遠点になっている。ここで、最近点のボロノイ図と最遠点のボロノイ図の結び (Union) をとると (図 4)、各結びの領域内の任意の点は最近点と最遠点が唯一に決められていることになる。

このことから、真円度を決定する同心円の中心は与えられた n 個の点に対する 2 つのボロノイ図の結びの領域 (凸



———— the nearest-point Voronoi diagram

----- the farthest-point Voronoi diagram

図 4 最近点のボロノイ図と最遠点のボロノイ図の結び

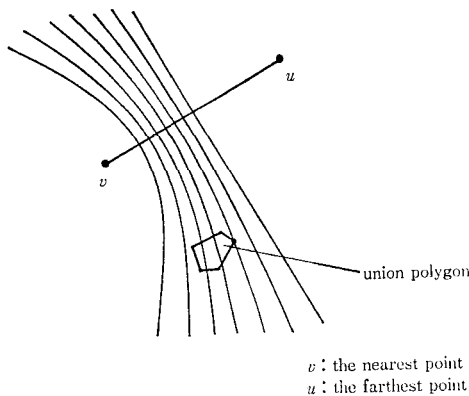


図 5 距離の差の関数と凸領域

領域となる)について,その各領域内の点の中で対応する最近点と最遠点の距離の差を最小とする領域内の点を見つければよいこととなる。

ここで,距離の差の関数は双曲線となるので,結びの領域側(最近点側)からみると凹関数であり,結びの領域は凸であるので厳密解は必ず結びの領域の頂点に存在する(図 5)。すなわち,真円度を決定する同心円の中心は,最近点のポロノイ点,最遠点のポロノイ点,あるいは,最近点のポロノイ辺と最遠点のポロノイ辺の交点に存在する。しかし,実際には最近点のポロノイ点と最遠点のポロノイ点には厳密解が存在しないことが証明できるので,ポロノイ辺の交点のみを調べればよいことになる[4]。

以上の論拠にもとづいた真円度を求めるアルゴリズムを「アルゴリズム 1」として次に述べる。

[アルゴリズム 1]

- step 1 最近点のポロノイ図を構成する。
- step 2 最遠点のポロノイ図を構成する。
- step 3 step 1, 2 で求めた 2 つのポロノイ図の結び (Union) をとる。
- step 4 2 つのポロノイ図の結びの図の中で,最近点のポロノイ辺と最遠点のポロノイ辺との交点に対して,その点からの最近点と最遠点の距離の差を求める。
- step 5 最近点と最遠点の距離の差が最小となる点とその値を求める。最小となる距離の差が真円度であり,最小となる点が真円度を決定する同心円の中心である。

[アルゴリズム 1] の時間計算量を考える。step 1, 2 は $O(n \log n)$ あるいは $O(n^2)$ である。step 3, 4, 5 では,

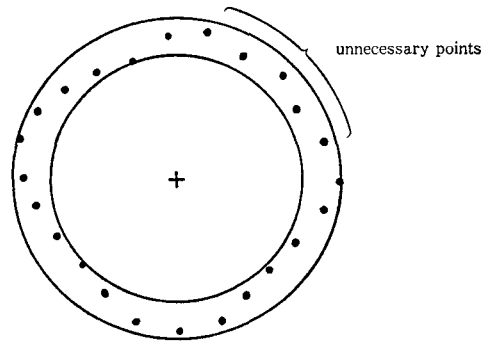


図 6 真円度問題における不要点

ポロノイ辺の交点を求めている。ポロノイ辺の数はおのおの $O(n)$ であり,その交点の数はたかだか $O(n^2)$ である。したがって,全体の時間計算量は $O(n^2)$ となる。ここで,ポロノイ辺の交点の数が $O(n^2)$ となる例がすでに知られている[4]。

3. 改訂真円度アルゴリズム

3.1 不要点の削除

実際の真円度測定においては実時間性も要求され,真円度の計算は 30 秒以内(遅くとも 1 分以内)が望まれる。しかし,データ点の数は 1000 点以上であり,いくら多項式時間でも $O(n^2)$ では十分な速度が得られず,高速化が必要である。

ここで,もう一度真円度問題を考えてみる。与えられた n 個の点の中には明らかに不要であると考えられる点(外接円あるいは内接円に関与しない点)がたくさん存在する(図 6)。これらの点(不要点)を厳密性を保証したままあらかじめ削除することができれば,高速化が図れると考える。

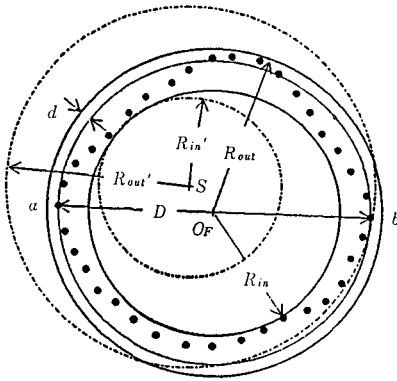
具体的には,まず,与えられた点に対して内接円に関与しない点の削除を行ない,これらの点に対して最近点のポロノイ図を構成する。次に,もとの与えられた点に対して外接円に関与しない点の削除を行ない,これらの点に対して最遠点のポロノイ図を構成する。そして,最後にこれら 2 つのポロノイ図の結びをとり,その交点から真円度を求める。

次に,厳密性を保証するための定理とアルゴリズムを述べる。

3.1.1 内接円に関する不要点削除

最遠点対を用いた内接円の半径の下界に関する定理を以下に示す。

[定理 1] データ点集合の最遠点対の中点を O_F , 最遠



point a, b : the farthest pair
 point O_F : the center of the farthest pair
 $d = R_{out} - \frac{D}{2}$

図 7 [定理 1] の説明図

点对の距離を D とする。点 O_F から最も近いデータ点までの距離を R_{in} , 最も遠いデータ点までの距離を R_{out} とすると, $R_{in} - (R_{out} - \frac{D}{2})$ は真円度の厳密解を与える内接円の半径の下界となる(図 7)。

[証明]

$r = \frac{D}{2}$, $d = R_{out} - r$ とする。ここで, $R_{out} - R_{in}$ は真円度の上界を示すことに注意しておく。このとき, $R'_{in} < R_{in} - d$ なる内接円の半径を持ち, かつ $R_{out} - R_{in}$ 未満の真円度を与えるような真円度の中心は存在しないことを以下に示す。

$$R_{out} - R_{in} = (R_{out} - d) - (R_{in} - d) = r - (R_{in} - d)$$

また, $R_{in} - d$ より半径の小さい半径 R'_{in} を持つ内接円の中心 S が存在するとする。そして, 点 S を中心とする外接円の半径を R'_{out} とする。

$2r = D$ は最遠点对間の距離を示しているのだから,

$$r \leq (\text{最小外接円の半径})$$

よって,

$$R'_{out} > r$$

である。また, 仮定により,

$$R'_{in} < R_{in} - d$$

であるので,

$$R_{out} - R_{in} = r - (R_{in} - d) < R'_{out} - R'_{in}$$

である。[証明終了]

[定理 1] から厳密性を保証した次のような内接円に関する不要点削除アルゴリズムを導くことができる([アルゴリズム 2])。このアルゴリズムは R. L. Graham の凸法アルゴリズム[5] と類似の形で形成されている。

ここで, データ点は半時間回りにポイントでつながれ

ているものとする。[アルゴリズム 2] では, 順方向のポイントを $L[v]$, 逆方向のポイントを $R[v]$ で表わす。

[アルゴリズム 2]

step 1 データ点集合の最遠点对をキャリバー法により求める[13]。

step 2 最遠点对の中点を O_F とする。また, 最遠点对の距離を D とする。

step 3 点 O_F に最も近い点までの距離を R_{in} , 点 O_F に最も遠い点までの距離を R_{out} とする。

step 4 $R_L = R_{in} - (R_{out} - \frac{D}{2})$ とする。

step 5 最遠点对のどちらか一方を start とする。特に $R[start] = start$ とする。

step 6 $v := start$;

WHILE ($L[v] \neq start$)

IF ($v, L[v], L[L[v]]$ が左折れ) THEN

$r := (v, L[v], L[L[v]])$ の 3 点でできる円の半径);

IF ($r < R_L$) THEN

$L[v]$ を削除;

$v := R[v]$;

ELSE

$v := L[v]$;

END;

ELSE

$v := L[v]$;

END;

END;

3.1.2 外接円に関する不要点削除

最大内接円を用いた外接円の半径の上界に関する定理を以下に示す。

[定理 2] データ点集合の最大内接円の中心を O_L とし, 点 O_L から最も遠いデータ点までの距離を R_{out} とすると, R_{out} は真円度の厳密解を与える外接円の半径の上界となる(図 8)。

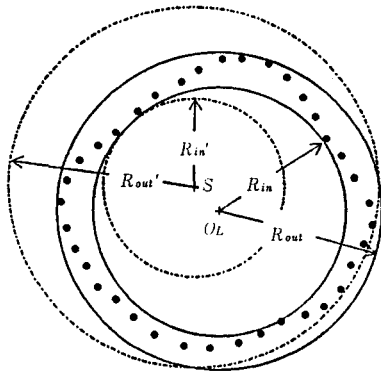
[証明]

最大内接円の半径を R_{in} とすると, $R_{out} - R_{in}$ は真円度の上界となることは明らかである。このとき, $R'_{out} > R_{out}$ なる外接円の半径を持ち, かつ $R_{out} - R_{in}$ 未満の真円度を与えるような真円度の中心は存在しないことを以下に示す。

R_{out} より大きい R'_{out} の半径の外接円を持つ中心 S が存在し, 点 S を中心とする内接円の半径を R'_{in} とする。

R_{in} は最大内接円の半径であるから,

$$R'_{in} < R_{in}$$



point O_L : the center of the largest empty circle

図 8 [定理 2] の説明図

である。また、仮定により、

$$R'_{out} > R_{out}$$

であるので、

$$R'_{out} - R'_{in} > R_{out} - R_{in}$$

である。[証明終了]

[定理 2] から厳密性を保証した次のような外接円に関する不要点削除アルゴリズムを導くことができる([アルゴリズム 3])。ここで、データ点のつながりは [アルゴリズム 2] と同様である。

[アルゴリズム 3]

- step 1 データ点集合の最大内接円の中心 O_L を求める。
 step 2 点 O_L から最も遠いデータ点までの距離を R_U とする。
 step 3 点 O_L から最も遠いデータ点を start とする。特に、 $R[start]=start$ とする。
 step 4 $v:=start$;
 WHILE ($L[v] \neq start$)
 $r := (v, L[v], L[L[v]])$ の 3 点でできる円の半径);
 IF ($r > R_U$) THEN
 $L[v]$ を削除;
 $v := R[v]$;
 ELSE
 $v := L[v]$;
 END;
 END;

3.2 改訂真円度アルゴリズム

以上の考察から、2つの不要点削除アルゴリズムを組み込んだ改訂真円度アルゴリズムを [アルゴリズム 4] として述べる。

[アルゴリズム 4]

step 1 [内接円に関する不要点削除]を行なう。

step 2 step 1 で残った点集合で最近点のポロノイ図を構成する。

step 3 最近点のポロノイ図を利用して、[外接円に関する不要点削除]を行なう。

step 4 step 3 で残った点集合で最遠点のポロノイ図を構成する。

step 5 step 2 で構成した最近点のポロノイ図と step 4 で構成した最遠点のポロノイ図の結びをとる。

step 6 2つのポロノイ図の結びの図の中で最近点のポロノイ辺と最遠点のポロノイ辺との交点に対して、その点からの最近点と最遠点の距離の差を求める。

step 7 最近点と最遠点の距離の差が最小となる点とその値を求める。最小となる距離の差が真円度であり最小となる点が真円度を決定する同心円の中心である

4. 計算機実験

実際の真円度測定に対して、[改訂真円度アルゴリズム]を適用した場合の効果を検証するために計算機上で実験を行なう。

本実験で用いたデータは、実際の真円度測定機から得られたもので、約 3 cm の円筒状断面より原点を中心とし反時計回りに 0.2 度ごとに 1800 点サンプリングしたものである。また、データの半径方向の偏差は数 μm 程度である。実際に用いた計算機は、CPU として MC68020 をもったソニー製の UNIX ワークステーションであり、使用言語は C 言語である。

表 1 に、凸包上のデータ点数とそれぞれの不要点削除によって残ったデータ点数を示す。

表 2 には、[アルゴリズム 1] と [アルゴリズム 4] の計算時間の比較を示す。

実験結果から、不要点削除を行なえばそれぞれのポロノイ図を構成するのに必要なデータ点の数は、最近点のポロノイ図の構成で約 20 分の 1、最遠点のポロノイ図の構成で約 200 分の 1 で済むことがわかる。また、計算時間では 20 倍以上高速になっている。ここで、[内接円に関する不要点削除]で削除されるデータ点の数が少ないのは、内接円の半径の下界が最遠点対から求められており、[外接円に関する不要点削除]での外接円の半径の上界に比べて良くないためであると考えられる。

5. むすび

本稿では、著者らのアルゴリズムを中心に真円度問題

表 1 各不要点削除後に残るデータ点の数

データ No.	凸包上のデータ点数	不要点削除後に残ったデータ数	
		内接円	外接円
1	1273	107	5
2	1297	114	12
3	1273	102	7
4	1254	111	13
5	1233	22	4
6	1201	112	7
7	1170	33	3
8	1630	9	4
平均	1291	76.3	6.9

について計算幾何学と関連づけながら概説した。少し理論的になってしまったのではないかと危惧している。

本稿では述べななかったが、真円度問題に関して別の定式化が提案されており、理論的に面白い結果が出ている[9]。また、詳しいことはまだ良く知らないが、この真円度問題に対する $O(n^5)$ アルゴリズムが提案されたという噂である。まだただ興味深い問題のようである。

最後に、本稿の解法に関してご指導いただいた大阪大学名誉教授(現在、国立奈良工業高等専門学校校長)中西義郎先生と、大阪大学工学部助教授中野秀男先生に感謝の意を表す。また、本稿の解法に関して共同研究者であった現在富士通株式会社の福山訓行氏に感謝する。

参 考 文 献

- [1] 浅野哲夫：“計算幾何学”，朝倉書店，1990年。
- [2] H. Ebara, N. Fukuyama, H. Nakano, and Y. Nakanishi：“A Practical Algorithm for Computing the Roundness”，Trans. IEICE, to appear.
- [3] 榎原博之，中野秀男，中西義郎，真田友宏：“ポロノイ図を応用した真円度を求める解法”，信学論，Vol. J70-A, No.4, pp.620-624, 1987年。
- [4] 福山訓行：“ポロノイ図を用いた真円度計算に関する研究”，大阪大学工学部通信工学科，修士論文，1988年。
- [5] R. L. Graham：“An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set”，Information Processing Letters, vol.1, pp.132-133, 1972.
- [6] P. J. Green and R. Sibson：“Computing Dirichlet Tessellations in the Plane”，The

表 2 [アルゴリズム 1] と [アルゴリズム 4] の計算時間の比較

データ No.	計算時間 (秒)	
	[アルゴリズム 1]	[アルゴリズム 2]
1	116.2	6.26
2	142.3	7.00
3	156.5	6.41
4	138.4	6.71
5	159.1	4.86
6	177.9	9.39
7	141.3	7.06
8	165.5	7.89
平均	149.7	6.95

Computer Journal, vol.21, no.2, pp.168-173, 1978.

- [7] 伊理正夫監修/塚塚武志編集：“計算幾何学と情報処理”，bit 別冊，共立出版，1986年。
- [8] ISO R1101/I, 1969.
- [9] V.B. Le and D.T. Lee：“Out-Of-Roundness Problem Revisited”，IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.13, no. 3, pp.217-223, 1991.
- [10] T.S.R. Murshy and S.Z. Abidin：“Minimum Zone Evaluation of Surfaces”，International Journal of Machine Tool Design & Research, vol.20, pp.123-137, 1980.
- [11] F. P. Preparata and M. I. Shamos：“Computational Geometry - An Introduction-”，Springer-Verlag, 1985.
- [12] M. I. Shamos and D. Hoey：“Closest-Point Problems”，Proceedings of 16th Annual IEEE Symposium on Foundations of Computer Science, pp.151-162, 1975.
- [13] G. T. Toussaint：“Solving Geometric Problems with the Rotating Calipers”，Proceedings of IEEE MELECON '83, 1983.
- [14] T. Tsukada and T. Kanada：“Measurement of Cylindrical Form Errors Using a Non-Contact Detector”，Precision Engineering, vol.4, pp.153-158, 1982.
- [15] 塚田忠夫，金田徹，奥田謙造：“最適化技法を用いた最小領域法真円度の評価法”，精密機械，Vol.49, No.10, pp.1351-1357, 1983年。