

## 近似計算による代数的数の符号判定

日吉 久礎    杉原 厚吉

東京大学大学院工学系研究科計数工学専攻

**概要** 実代数的数を係数とする多項式  $f(x_1, \dots, x_n)$  および実代数的数  $\alpha_1, \dots, \alpha_n$  に対して,  $f(\alpha_1, \dots, \alpha_n)$  の符号を判定する問題を考察する. 従来のアプローチは,  $f(\alpha_1, \dots, \alpha_n)$  を厳密計算によって求めて符号を判定するというものであった. これに対し, 関川 [5] は, 機械区間演算と Mahler の measure を組み合わせた実代数的数の符号判定法を提案した. しかし, 関川の符号判定法においては, 符号判定の可能な精度を事前に知ることができなかった. 本稿では, 符号判定の可能な精度を事前に推定し, かつ機械区間演算の代わりに浮動小数点演算を用いる実代数的数の符号判定法を提案する. また, 2 つの方法を凸包を構成するための Graham のアルゴリズムに対して応用し, 比較を行なう.

## A Method for Deciding the Signs of Algebraic Numbers Using Approximate Arithmetic

Hisamoto Hiyoshi and Kokichi Sugihara

Department of Mathematical Engineering and Information Physics,  
Graduate School of Engineering, University of Tokyo

**abstract** This paper considers the problem of deciding the sign of  $f(\alpha_1, \dots, \alpha_n)$  where  $f(x_1, \dots, x_n)$  is a given polynomial whose coefficients are real algebraic numbers and  $\alpha_1, \dots, \alpha_n$  are given real algebraic numbers. Approaches known so far are to decide the sign of  $f(\alpha_1, \dots, \alpha_n)$  using exact operations. Recently, Sekigawa [5] combined machine interval arithmetic with Mahler's measure and proposed a new method using approximation techniques. In his method, however, the required precision cannot be known in advance. The Method proposed in this paper, on the other hand, estimates the required precision in advance and uses floating point arithmetic instead of machine interval arithmetic. The proposed method and Sekigawa's method are compared experimentally for Graham's algorithm for constructing convex hulls.

## 1 はじめに

実係数多項式  $f(x_1, \dots, x_n) \in \mathbf{R}[x_1, \dots, x_n]$  に実数  $a_1, \dots, a_n$  を代入したときの値の符号判定は、計算代数や計算幾何における多くのアルゴリズムで用いられる。例えば、2次元凸包を構成する Graham のアルゴリズム [2] では、3点  $p_i = (x_i, y_i), i = 1, 2, 3$ , の位置関係を判定するが、この判定は通常

$$\Delta(p_1, p_2, p_3) = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} \quad (1)$$

の値の符号を求めることで行なわれる。

本研究では、上の符号判定問題を実代数的数に制限する。以下では、代数的数の集合を  $\mathbf{K}$  と書き、実代数的数の集合を  $\mathbf{A}$  と書くことにする。本研究で考察するのは次の問題である。

**問題** 多項式  $f(x_1, \dots, x_n) \in \mathbf{A}[x_1, \dots, x_n]$  および実代数的数  $\alpha_1, \dots, \alpha_n \in \mathbf{A}$  が与えられたとする。このとき、 $f(\alpha_1, \dots, \alpha_n)$  の符号を求めよ。

この問題に対する従来のアプローチは、厳密計算によって  $f(\alpha_1, \dots, \alpha_n)$  の値を実際に計算し、その符号を求めるというものであった [3]。例えば、一般に、任意の実代数的数  $\alpha$  は、 $\alpha$  を根として持つ多項式  $P$  および  $\alpha$  を含み他の  $P$  の根を含まない区間  $I$  の組  $(P, I)$  で表現することができる。この表現法による場合、実代数的数の間で演算した結果の最小多項式を求めるために、終結式の計算などの多大なコストを要する計算を行なうことになる。ところが、今考えている問題は、 $f(\alpha_1, \dots, \alpha_n)$  の値を求める問題ではなく、その符号を求める問題である。したがって、計算の途中結果を厳密に表現する必要はなく、符号判定に必要な情報だけを計算するだけでよい。このような考え方に基づく符号判定法の1つとして、関川の符号判定法 [5] がある。関川の符号判定法は、Mahler の measure [1] と機械区間演算 [4] を用いた、近似計算によるものである。

関川の符号判定法では、まずある精度  $\mu$  の機械区間演算を用いて  $f = f(\alpha_1, \dots, \alpha_n)$  の符号判定を試みるが、必ずしも  $f$  の符号が確定するとは限らない。そこで、精度  $\mu$  をあげて再度符号判定を試みるということを繰り返す。すなわち、 $f(\alpha_1, \dots, \alpha_n)$  の値を、近似計算ではあるが、精度を変えて何度も計算する可能性がある。本研究では、符号判定を行なうときに機械区間演算を用いる代わりに通常の浮動小数点演算を用い、さらに  $f$  の符号判定の可能な精度を前もって求めておき、 $f(\alpha_1, \dots, \alpha_n)$  の値の計算を1度で済ませることのできるような符号判定法を提案する。

2節で、Mahler の measure を概説する。3節で、関川の符号判定法を概説する。4節で、本研究で提案する符号判定法を概説する。5節で、関川の符号判定法および本研究で提案する符号判定法を Graham のアルゴリズムに応用し比較する実験を行なう。

## 2 Mahler の measure

代数的数の measure [1] を以下のように定義する。

**定義 1** 代数的数  $\alpha \in \mathbf{K}$  に対して、 $\alpha$  の整数係数の原始最小多項式  $g(x) \in \mathbf{Z}[x]$  を

$$g(x) = a_d x^d + \dots + a_0 = a_d \prod_{i=1}^d (x - \beta_i)$$

と書いたとき、 $\alpha$  の measure を

$$M(\alpha) = |a_d| \prod_{i=1}^d \max\{1, |\beta_i|\}$$

と定義する。

代数的数の measure と絶対値の間に次の補題が成り立つ。

**補題 2** 任意の0でない代数的数  $\alpha \in \mathbf{K}$  に対して、

$$1/M(\alpha) \leq |\alpha| \leq M(\alpha)$$

が成り立つ。  $\square$

補題 2 は, measure に基づいた実代数的数の符号判定の基礎を与える (図 1 参照).

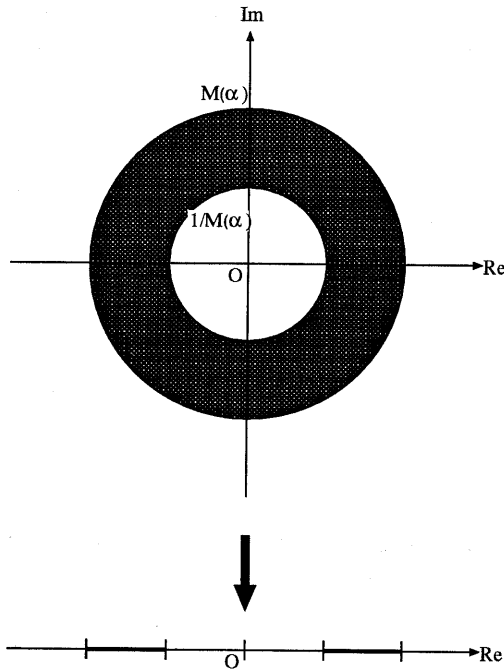


図 1 代数的数  $\alpha \neq 0$  の存在範囲 (網かけ部)

代数的数の measure を厳密にあるいは近似的に計算する方法にはいくつかある [1]. 次の補題は代数的数  $\alpha \in \mathbf{K}$  の最小多項式のノルムによって  $\alpha$  の measure を上から評価する方法を与える.

補題 3 任意の複素係数多項式

$$g(x) = a_d x^d + \cdots + a_0 = a_d \prod_{i=1}^d (x - \beta_i) \in \mathbf{C}[x]$$

に対して,

$$|a_d| \prod_{i=1}^d \max\{1, |\beta_i|\} \leq \|g\|_2$$

が成り立つ. ここで,  $\|g\|_2$  は

$$\|g\|_2 := \sqrt{|a_d|^2 + \cdots + |a_0|^2}$$

と定義される. □

measure に基づいた実代数的数の符号判定においては,  $f(\alpha_1, \dots, \alpha_n)$  の measure の大きさを上から評価しなければならないが, 次の補題はその基礎を与える.

補題 4 任意の代数的数  $\alpha, \beta \in \mathbf{K}$  に対して,  $\alpha, \beta$  の次数をそれぞれ  $d, e$  としたとき,

$$M(\alpha \pm \beta) \leq 2^{de} M(\alpha)^e M(\beta)^d$$

$$M(\alpha\beta) \leq M(\alpha)^e M(\beta)^d$$

が成り立つ. □

### 3 関川の符号判定法

この節では, 精度  $\mu$  によって指定される機械区間 [4] の集合を  $\mathbf{IM}_\mu$  と書き,  $\mathbf{IM}_\mu$  上の機械区間演算を  $+_\mu, -_\mu, \cdot_\mu$  と書く.

関川 [5] は, 機械区間演算と measure を組み合わせた実代数的数の符号判定法を提案した. 多項式  $f(x_1, \dots, x_n) \in \mathbf{A}[x_1, \dots, x_n]$  および実代数的数  $\alpha_1, \dots, \alpha_n \in \mathbf{A}$  が与えられたとする. 関川の符号判定法では, 実代数的数  $\alpha$  を厳密に表現する代わりに, 組  $(I, d, M)$  によって表す. ここで,  $I$  は  $\alpha$  を含む区間,  $d$  は  $\alpha$  の次数の上界,  $M$  は  $\alpha$  の Mahler の measure の上界である. 組  $(I, d, M)$  を  $\alpha$  の代数的情報つき区間と呼ぶ.  $I \in \mathbf{IM}_\mu$  のとき  $(I, d, M)$  を精度  $\mu$  の代数的情報つき区間と呼ぶ.

定義 5 精度  $\mu$  の代数的情報つき区間上の演算を次のように定義する:

$$(I, d, M) + (J, e, N) = (I +_\mu J, de, 2^{de} M^e N^d),$$

$$(I, d, M) - (J, e, N) = (I -_\mu J, de, 2^{de} M^e N^d),$$

$$(I, d, M) \cdot (J, e, N) = (I \cdot_\mu J, de, M^e N^d).$$

アルゴリズム 6 与えられた多項式  $f(x_1, \dots, x_n) \in A[x_1, \dots, x_n]$  および実代数的数  $\alpha_1, \dots, \alpha_n \in A$  に対して,  $f(\alpha_1, \dots, \alpha_n)$  の符号を求める.

Input: 多項式  $f \in A[x_1, \dots, x_n]$ ,  
実代数的数  $\alpha_1, \dots, \alpha_n$ .

Output:  $\text{sgn}(f(\alpha_1, \dots, \alpha_n))$ .

- 1  $\mu$  を設定する.
- 2  $f$  の係数および  $\alpha_1, \dots, \alpha_n$  を精度  $\mu$  の代数的情報つき区間に変換する.
- 3 代数的情報つき区間の間で演算をする. この結果を  $(I, d, M)$  とする.
- 4 if  $I > 0$  then return 1;
- 5 if  $I < 0$  then return -1;
- 6 return 0;
- 7 精度  $\mu$  を上げる.
- 8 goto 2;

#### 4 提案する符号判定法

関川の符号判定法では,  $f(\alpha_1, \dots, \alpha_n)$  の符号が判定できるまで精度  $\mu$  をあげていった. この節では, 符号判定の可能な精度を事前に推定し, また, 区間演算を用いなくて符号を判定する方法を提案する. 提案する符号判定法は, 実代数的数を浮動小数点数によって近似し,  $f(\alpha_1, \dots, \alpha_n)$  を近似的に計算した結果と  $f(\alpha_1, \dots, \alpha_n)$  の measure に関する情報から符号を決定するというものである.

浮動小数点体系は, 基数, 桁数, 最小指数および最大指数によって特定される. 本研究では, 基数  $\beta = 2$ , 桁数  $\mu$  の浮動小数点体系を用いる. また, 十分な広い範囲の数を扱えるという意味で, 最小指数  $L = -\infty$ , 最大指数  $M = +\infty$  と仮定する (注意 7 参照). したがって, 本研究で用いる浮動小数点体系は, 桁数のパラメータ  $\mu$  のみで指定される. 以下では,  $\mu$  を

浮動小数点体系の精度と呼ぶことにし, 精度  $\mu$  の浮動小数点体系で表現することのできる数値の集合を  $M_\mu \subset \mathbb{R}$  と書くことにする. また, 本研究では, 丸めの方式として切捨てを採用する. 精度  $\mu$  の浮動小数点体系の加減乗算を  $+, -, \cdot_\mu$  と書く. 多項式  $f$  に対して, 加減乗算を対応する浮動小数点演算にそれぞれ置き換えたものを  $f_\mu$  と書く.

注意 7 通常, 計算機に浮動小数点体系を実装するとき, 指数を固定の長さで表現することが多い. この場合,  $L = -\infty, M = +\infty$  という仮定は満たさず, オーバーフローやアンダーフローが発生することがありうる. したがって, 指数部を表現する長さは十分な大きさを持っていなければならない. この点について, さらに注意 13 を参照.  $\square$

次の定理が, 提案する符号判定法の基礎を与える.

定理 8 実代数的数  $\alpha^* \in A$  および浮動小数点数  $\alpha_\mu \in M_\mu$  に対して,

$$|\alpha_\mu - \alpha^*| < 2^{\xi - \mu}, M(\alpha^*) < 2^m \quad (2)$$

が成り立っていると仮定する. このとき,

$$\mu > \xi + m + 1 \quad (3)$$

が成り立っているならば,  $\alpha^*$  の符号が確定する. すなわち,

- (i)  $\alpha_\mu > 2^{-m-1}$  ならば  $\alpha^* > 0$ .
- (ii)  $\alpha_\mu < 2^{-m-1}$  ならば  $\alpha^* < 0$ .
- (iii)  $|\alpha_\mu| \leq 2^{-m-1}$  ならば,  $\alpha^* = 0$ .

証明 (3) から,

$$|\alpha_\mu - \alpha^*| < 2^{-m-1} \quad (4)$$

が導かれる. したがって, 以下では (4) を仮定する. 今,  $|\alpha_\mu| \leq 2^{-m-1}$  と仮定すると,

$$|\alpha^*| \leq |\alpha_\mu| + |\alpha_\mu - \alpha^*| < 2^{-m}$$

であるから、補題 2 より  $\alpha^* = 0$ . また、 $|\alpha_\mu| > 2^{-m-1}$  とすると、(4) から  $\alpha^* = 0$  が成り立つ。したがって、(4) の仮定においては、 $\alpha^* = 0$  と  $|\alpha_\mu| \leq 2^{-m-1}$  は同値である。

残りの場合についても同様に示せる。 □

#### 4.1 誤差解析

定理 8 より、 $\alpha^* = f(\alpha_1, \dots, \alpha_n) \in \mathbf{A}$  に対して、精度  $\mu$  の浮動小数点体系を用いて  $\alpha^*$  の近似  $\alpha_\mu$  が得られたとき、(2)(3) を満たすならば、 $\alpha_\mu$  の値から  $\alpha^*$  の符号を得ることができる。この節では、(2)(3) を満たす精度  $\mu$  を推定する方法の基礎を与え、4.2 節で  $\mu$  の推定法を示す。

補題 9 実数  $x^* \in \mathbf{R}$  を浮動小数点数  $x_\mu \in \mathbf{M}_\mu$  に丸めたととき、 $|x^*| < 2^a$  ならば、

$$|x_\mu| < 2^a, |x_\mu - x^*| < 2^{a-\mu}$$

である。 □

補題 10 実数  $x^*, y^* \in \mathbf{R}$  とその近似  $x_\mu, y_\mu \in \mathbf{M}_\mu$  が、

$$\begin{cases} |x_\mu| < 2^a, |x_\mu - x^*| < 2^{\xi-\mu}, \\ |y_\mu| < 2^b, |y_\mu - y^*| < 2^{\eta-\mu} \end{cases}$$

を満たすとす。このとき、

$$\begin{cases} |x_\mu \pm y_\mu| < 2^{e_1}, \\ |x_\mu \cdot y_\mu| < 2^{e_2}, \\ |(x_\mu \pm y_\mu) - (x^* \pm y^*)| < 2^{e_3-\mu}, \\ |x_\mu \cdot y_\mu - x^* \cdot y^*| < 2^{e_4-\mu} \end{cases}$$

が成り立つ。ここで、

$$\begin{aligned} e_1 &= \max\{a, b\} + 1, \\ e_2 &= a + b, \\ e_3 &= \max\{a, b, \xi, \eta\} + 2, \\ e_4 &= \max\{a + b, a + \eta, b + \xi, \xi + \eta\} + 2 \end{aligned}$$

である。

証明 絶対値の見積りについては自明。加減算の誤差については、

$$\begin{aligned} & |(x_\mu \pm y_\mu) - (x^* \pm y^*)| \\ & \leq |(x_\mu \pm y_\mu) - (x_\mu \pm y_\mu)| \\ & \quad + |x_\mu - x^*| + |y_\mu - y^*| \\ & < 2^{\max\{a, b\} + 1 - \mu} + 2^{\xi - \mu} + 2^{\eta - \mu} \\ & \leq 2^{\max\{a, b, \xi, \eta\} + 2 - \mu}, \end{aligned}$$

乗算の誤差については、

$$\begin{aligned} & |x_\mu \cdot y_\mu - x^* \cdot y^*| \\ & \leq |x_\mu \cdot y_\mu - x^* \cdot y^*| + |x_\mu \cdot y_\mu - x^* \cdot y^*| \\ & \leq |x_\mu \cdot y_\mu - x^* \cdot y^*| + |x_\mu - x^*| |y_\mu| \\ & \quad + |y_\mu - y^*| |x_\mu| + |x_\mu - x^*| |y_\mu - y^*| \\ & < 2^{a+b-\mu} + 2^{a+\eta-\mu} + 2^{b+\xi-\mu} + 2^{\xi+\eta-2\mu} \\ & < 2^{a+b-\mu} + 2^{a+\eta-\mu} + 2^{b+\xi-\mu} + 2^{\xi+\eta-\mu} \\ & \leq 2^{\max\{a+b, a+\eta, b+\xi, \xi+\eta\} + 2 - \mu} \end{aligned}$$

のように導かれる。 □

#### 4.2 浮動小数点演算による符号判定法

この節では、実代数的数の符号判定を用いるアルゴリズムを実装する際プログラムが扱う多項式  $f \in \mathbf{A}[x_1, \dots, x_n]$  の係数と実代数的数  $\alpha_1, \dots, \alpha_n \in \mathbf{A}$  の範囲がわかっている場合に、(2)(3) を満たす精度  $\mu$  を事前に計算しておく方法を提案する。  $f$  の係数および  $\alpha_1, \dots, \alpha_n$  としてとりうる実代数的数の集合を  $A$  と書く。

(i) 任意の  $\alpha^* \in A$  に対して

$$M(\alpha^*) < 2^m, \deg(\alpha^*) \leq d$$

を満たすような  $m$  および  $d$  を求める。これは補題 3 などを用いて求めることができる。

(ii) 精度  $\mu$  の浮動小数点体系を用いると仮定して、任意の  $\alpha^* \in A$  に対して、

$$|\alpha_\mu| < 2^a, |\alpha_\mu - \alpha^*| < 2^{\xi-\mu},$$

を満たすような  $a$  および  $\xi$  を求める。これは補題 9 から求めることができる。

(iii) 以上の評価と補題 4, 10 を用いて、任意の  $g \in A[x_1, \dots, x_n]$  および  $\beta_1, \dots, \beta_n \in A$  に対する  $g(\beta_1, \dots, \beta_n)$  の measure および誤差の上界を計算する。

(iv) (3) から、任意の  $g \in A[x_1, \dots, x_n]$  および  $\beta_1, \dots, \beta_n \in A$  に対して  $g(\beta_1, \dots, \beta_n)$  の符号判定の可能な精度  $\mu$  を求めることができる。

上の方法を例を用いて説明する。

例 11 多項式 (1) および有理数の集合

$$Q = \{p/q \mid 0 < p < q < 2^M\}, \quad M \in \mathbf{Z}_{>0}$$

を考える。このとき、任意の  $q^* \in Q$  に対して、

$$M(q^*) < 2^M, 0 \leq |q_\mu| < 2^0, |q_\mu - q^*| < 2^{-\mu}$$

が成り立つ。したがって、補題 4, 10 より、任意の  $x_1, \dots, y_3 \in Q$  に対して、

$$M(\Delta(x_1, \dots, y_3)) < 2^{8M+7}$$

$$|\Delta_\mu(x_{1\mu}, \dots, y_{3\mu}) - \Delta(x_1, \dots, y_3)| < 2^{8-\mu}$$

が成り立つ。定理 8 より、符号が確定するための十分条件は

$$2^{8-\mu} < \frac{1}{2^{8M+8}}$$

であることがわかる。これを解いて、

$$\mu > 8M + 16$$

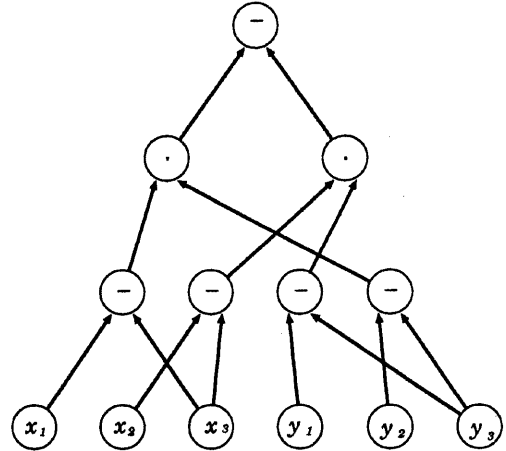
を得る (図 2 参照)。 □

例 12 多項式 (1) および集合

$$Q = \{\sqrt{p/q} \mid 0 < p < q < 2^M\}, \quad M \in \mathbf{Z}_{>0}$$

を考える。このとき、任意の  $q \in Q$  に対して、

$$M(q) < 2^M, 0 \leq |q| < 2^0, |q_\mu - q| < 2^{-\mu}$$



|      | measure の上界 | 誤差の上界       |
|------|-------------|-------------|
| 1 段目 | $2^M$       | $2^{-\mu}$  |
| 2 段目 | $2^{2M+1}$  | $2^{2-\mu}$ |
| 3 段目 | $2^{4M+3}$  | $2^{6-\mu}$ |
| 4 段目 | $2^{8M+7}$  | $2^{8-\mu}$ |

図 2 精度の事前推定

が成り立つ。したがって、補題 4, 10 より、任意の  $x_1, \dots, y_3 \in Q$  に対して、

$$M(\Delta(x_1, \dots, y_3)) < 2^{1024M+1792}$$

$$|\Delta_\mu(x_{1\mu}, \dots, y_{3\mu}) - \Delta(x_1, \dots, y_3)| < 2^{8-\mu}$$

が成り立つ。例 11 と同様にして、

$$\mu > 1026M + 1801 \quad (5)$$

を得る。 □

注意 13 注意 7 では、通常の浮動小数点体系の実装では、仮定  $L = -\infty, M = +\infty$  が満たされないことを指摘した。したがって、オーバーフロー、アンダーフローが発生することがありうるということに注意しなければならない。

この事態は次のようにして回避できる: 0 でない実代数的数  $\alpha \in A$  の絶対値は、補題 2 によって見積もることができる。したがって、上の例のように、あ

りうる実代数的数の measure の大きさをあらかじめ見積もり, オーバーフロー, アンダーフローが発生しないように  $-L, M$  を十分大きくとればよい. □

## 5 計算実験

関川の符号判定法および本研究で提案する符号判定法を Graham のアルゴリズムに応用し, 計算機 (SUN SPARC station 5, 110MHz) に実装した. プログラム言語は C++ を使用した. 基数  $2^{32}$  の可変精度浮動小数点ライブラリを作成し, 使用した. 計算時間は, SUN OS version 4.1.3 の time コマンドが出力する user time によって測定した.

関川の符号判定においては, 初期精度および精度の上げ方が任意である. ここでは, 初期精度を 1 とし, 2 倍ずつ上げていった.

本研究で提案する符号判定法においては, 例 12 にしたがって符号判定の可能な精度を推定した. 例 12 の精度の推定は, 基数 2 の浮動小数点体系に対するものである. そのまま使用することはできないが, 簡単な考察により, (5) を満たす  $\mu$  に対して  $\lceil \mu/32 \rceil + 1$  桁の浮動小数点体系を用いれば十分であることがわかる. 最初, 精度 1 の機械区間演算を用いて符号判定し, 区間に 0 が含まれる場合に提案した符号判定法を用いるというスキームを用いた.

どちらの方法に対しても, 代数的数を  $\mu$  桁で近似する必要がある. 当初, 関川 [6] に従い, 区間縮小法によって代数的数の近似を行なった. しかし, 区間縮小法は 1 次収束なので, 多くの時間がかかり, 計算時間が日単位になってしまった. そこで, 代数的数の近似に区間 Newton 法 ([4] p. 66, (5.16)) を用いた.

実験 1 点の座標として,

$$\left( \sqrt{p/q}, \sqrt{r/s} \right),$$

$$q = 2, \dots, 255, \quad p = 1, \dots, q-1,$$

$$s = 2, \dots, 255, \quad r = 1, \dots, s-1,$$

の形の数をとったときの凸包を構成した (図 3).

実験 2 点の座標として,

$$\left( \sqrt{p/(q \pm 1)}, \sqrt{p/q} \right)$$

$$q = 129, \dots, 254, \quad p = 127, \dots, q-2$$

の形の数をとったときの凸包を構成した (図 4).

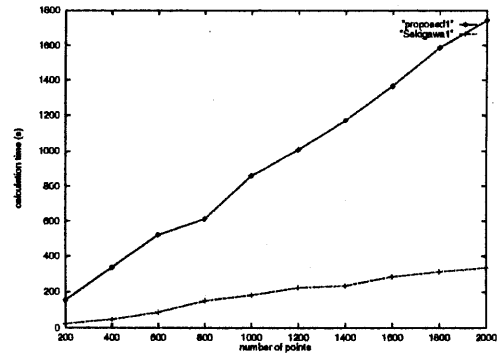


図 3 実験 1 の計算時間

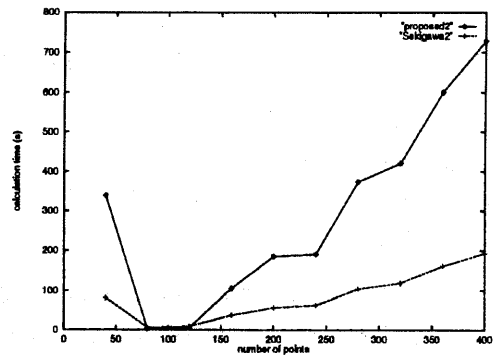


図 4 実験 2 の計算時間

## 6 結論

- 浮動小数点演算を用いた実代数的数の符号判定法を提案した。
- 関川の符号判定法および本研究で提案した符号判定法を Graham のアルゴリズムに応用・実装し、これらの方法によって符号判定ができることを確認した。
- 2 つの符号判定法を実験的に比較した。実験の結果、本研究で提案した符号判定法よりも、関川の符号判定法の方が高速であることがわかった。

## 7 今後の発展

- 代数的数を座標に持つ点の集合の凸包の工学的応用を研究する。
- 本研究で提案した精度推定法は、誤差や *measure* の評価式を帰納的にすることで精度推定を自動化することを念頭においたものである。そこで、精度推定をプログラム実行時に行なうプログラムを実装する。
- 提案した精度推定法よりも精密な精度推定法を開発し、高速化を目指す。
- Graham のアルゴリズム以外のアルゴリズムに対して、*measure* に基づく符号判定理論を応用する。
- 本研究で提案した符号判定法の計算量の理論的な解析をする。また、従来の厳密計算に基づく符号判定法と計算量を比較する。
- *measure* に基づく符号判定法は、最小多項式の代わりに *measure* を用いることで、代数的情報の計算を数値的情報の計算に置き換えたものである。したがって、代数的情報の計算と

数値的情報の計算の間にトレードオフがあると考えられる。そこで、最小多項式と *measure* の中間的な情報量を持つ代数的情報量を探し、より少ない計算量で符号判定のできる方法を開発する。

## 参考文献

- [1] L. Cerlienco, M. Mignotte, and F. Piras, Computing the measure of a polynomial, *J. Symbolic Computation* 4 (1987), pp. 21–33.
- [2] R. L. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Info. Proc. Lett.* 1 (1972), pp. 132–133.
- [3] R. Loos, Computing in algebraic extensions, B. Buchberger, G. E. Collins, and R. Loos eds., *Computer Algebra Symbolic and Algebraic Computation* (second edition), Springer-Verlag, Wien, 1983, pp. 173–187.
- [4] R. E. Moore, *Methods and Applications of Interval Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, 1979.
- [5] 関川 浩, 近似計算による代数的数の符号判定について, 京都大学数理解析研究所講究録 941 (1996), pp. 185–193.
- [6] 関川 浩, 区間演算と多項式ノルムによる代数的数の符号判定, 京都大学数理解析研究所講究録 *Researches on Algorithms for Computer Algebra*, preprint.