東邦大学大学院 理学研究科 情報科学専攻

2014年度 修士論文

# 安定化理論に基づくISCZ法の新しい実装方法の提案

指導教員：白柳 潔

提出日：2015年3月11日(水)

提出者：6513001 伊井 誠和

# 要 旨

一般的に計算機では小数はある桁で切り捨てを行い、近似計算を行なっている。近似計算では厳密計算に比べて誤差が生じやすく、アルゴリズムによっては全く違う結果を出力することがある。これを不安定なアルゴリズムと呼ぶ。この不安定なアルゴリズムを安定させるために、白柳らは安定化手法を提案した。

その安定化理論の応用である ISCZ 法では、区間とともに入力係数をシンボルとして保存し、区間演算を行う際には演算履歴をシンボルによって保存する。保存したシンボルはゼロ書き換えを行う際に実数値に復元し本当に 0 かどうかの評価を行う。

本研究では数式処理システム以外でも実装が可能となるような実装方法、およびゼロ書き換えの際のシンボルの評価手法を提案し、数式処理ソフト Maple を用いて実装を行った。実験には不安定なアルゴリズムであるユークリッドの互除法、スツルムアルゴリズム、グラハムアルゴリズムを用いた。その結果、無理数点を用いたグラハムアルゴリズムにおいては ISCZ 法が正確演算よりも高速に計算を終えられた。

また、スツルムアルゴリズムについては、数式処理システム以外の言語として JavaScript を用いて実験を行い、実際に提案手法が実装可能であることを確認した。

# 目 次

# 1　はじめに

一般的に計算機では小数はある桁で切り捨てを行い、近似計算を行なっている。近似計算では厳密計算に比べて誤差が生じやすく、アルゴリズムによっては全く違う結果を出力することがある。

[近似計算で誤った結果を出す例]

厳密計算

$$1-\frac{1}{3}\times 3=0 \text{ は真か?}$$

$$1-\frac{1}{3}\times 3=1-1=0$$

$$>YES$$

近似計算

$$1-\frac{1}{3}\times 3=0 \text{ は真か?}$$

$$1-\frac{1}{3}\times 3=1-0.333\times 3=0.001$$

$$>NO$$

このような厳密計算と近似計算で異なる結果を出しやすいアルゴリズムを、「不安定なアルゴリズム」と呼ぶ。この不安定なアルゴリズムを安定させるために、白柳らは安定化手法を提案した ([1],[2])。その安定化手法は、入力した多項式の係数を区間化して区間演算を行い、If 文のところでは、0 が含まれる区間を 0 に変換するという「ゼロ書き換え」を施すものである。

[安定化手法を用いた計算例]

厳密計算

$$1-\frac{1}{3}\times 3=0 \text{ は真か?}$$

$$1-\frac{1}{3}\times 3=1-1=0$$

$$>YES$$

安定化手法を用いた計算

$$1-\frac{1}{3}\times 3=0 \text{ は真か?}$$

$$\begin{aligned}1-\frac{1}{3}\times 3&=[1,1]-[0.333,0.334]\times[3,3]\\&=[-0.002,0.001]\\&=0\\&>YES\end{aligned}$$

安定化手法はすべてのアルゴリズムに適用可能ではなく、

- 入力が多項式である
- 加減乗除演算のみで構成されている
- 「YES か NO か」の判定が「0 であるか?」もしくは「正 (負) であるか?」

の 3 点全てを満たさなければならない。安定化手法の適用可能条件は厳しいが、適用出来るアルゴリズムでは近似計算よりも厳密計算に近い値を返し、厳密計算よりも短時間で解を求めることができる。

しかし、現在の研究では精度桁を増やすことで厳密解に近い解を返すことは保証されているが、精度桁が何桁以上のときに確実に安定するかは未解決問題となっている。この精度桁問題が解決すれば、大きな桁数で計算する必要がなくなるため、さらに早く正確な解を求めることが可能となる。安定化理論については 2 章で説明する。

前述の例にある安定化手法は区間演算とゼロ書き換えのみを用いているため IZ 法 (Interval method with zero rewriting) と呼んでいる。IZ 法では出力として区間を返すため、正確な答えが何であったかはわからない。そこで、入力実数と計算過程をシンボルとして保存し、ゼロ書き換えの際にシンボルを実数に戻し、本当に 0 であるかどうかを判定する ISCZ 法 (IS method with correct zero rewriting) が提案された。ISCZ 法については 3 章で説明する。

今回の研究では、今まで数式処理システムで実装されてきた ISCZ 法を用いたアルゴリズムを、その他の一般プログラミング言語でも実装できるように新しい実装方法を考えた。この方法については 5 章で説明する。

そして、数式処理ソフト Maple14 を用いて 3 種の ISCZ 法を用いたアルゴリズムを実装し実験した。そのうち 1 種については、数式処理システム以外の言語として JavaScript を用いて実験を行った。実験結果については 6 章で詳しく説明する。

## 2　安定化理論

次のアルゴリズムを対象に安定化理論の復習を簡単に行う。詳細はを参照されたい。

- データは、すべて多項式環 $R[x_1, \cdots, x_m]$ の元からなる。$R$ は実数体の部分体である。
- データ間の演算は、$R[x_1, \cdots, x_m]$ 内の加減乗除である。
- データ上の述語は、不連続点をもつとすればそれは 0 のみである。

述語の不連続点が 0 という意味は、If "$C = 0$" then ... else ... のように、値が 0 か否かによって分岐が別れることである。従って、$C = 0$ の代わりに $C > 0$ や $C \leq 0$ などでもよい。上記クラスのアルゴリズムを、不連続点 0 の代数的アルゴリズムと呼ぶ。ほとんどの数式処理のアルゴリズムはこのクラスに入るか、このクラスのアルゴリズムに変換可能である。

さて、安定化の 3 つのポイントは、

- アルゴリズムの構造は変えない。
- データ領域において、ふつうの係数を区間係数に変える。
- 述語の評価の直前で、区間係数のゼロ書換えを行なう。

である。すなわち、安定化されたアルゴリズムは次のようになる。

**区間領域** データ領域は区間係数多項式の集合。区間係数は $[A, B]$ なる形で、$A, B \in \mathbb{R}$, $[A, B]$ は集合 $\{r \in \mathbb{R} \mid A \leq r \leq B\}$ を意味する。

**区間演算** 二項演算 $\star \in \{+, -, \times, \div\}$ に対し、

$$[A, B] \star [C, D] = [min(A \star C, A \star D, B \star C, B \star D), max(A \star C, A \star D, B \star C, B \star D)]$$

**ゼロ書換え** 不連続点 0 をもつ述語を評価する直前で、各区間係数 $[A, B]$ に対し、

$A \leq 0 \leq B$ ならば $[A, B]$ を $[0, 0]$ に書き換えよ。
そうでないならばそのままとせよ。

今、入力 $f \in R[x_1, \cdots, x_m]$ を

$$f = \sum_{i_1, \cdots, i_m} r_{i1 \cdots im} x_1^{i_1} \cdots x_m^{i_m}$$

と表したとき、$f$ に対する近似列 $Int(f)_j$ を

$$Int(f)_j = \sum_{i_1, \cdots, i_m} [(a_{i1 \cdots im})_j, (b_{i1 \cdots im})_j] x_1^{i_1} \cdots x_m^{i_m}$$

で定義する。ここに、すべての $i_1, \cdots, i_m$ について、

$$(a_{i1} \cdots {}_{im})_j \leq r_{i1} \cdots {}_{im} \leq (b_{i1} \cdots {}_{im})_j \ for \ \forall j$$

$$(b_{i1} \cdots {}_{im})_j - (a_{i1} \cdots {}_{im})_j \to 0 \ as \ j \to \infty$$

このとき、単に

$$Int(f)_j \to f$$

と書く。

さて、A を安定化したアルゴリズムを $Stab(\mathcal{A})$ と書くと、次が安定化理論の基本定理 ([3]) である。

**定理 1 (安定化理論の基本定理)**
$A$ は不連続点 0 の代数的アルゴリズムで、入力 $f \in R[x_1, \cdots, x_m]$ に対し正常終了するとせよ。このとき、$f$ に対する任意の近似列 $\{Int(f)_j\}_j$ に対し、ある $n$ が存在して、$j \geq n$ ならば、$Stab(\mathcal{A})$ は $\{Int(f)_j\}_j$ に対し正常終了し、

$$Stab(\mathcal{A})(Int(f)_j) \to \mathcal{A}(f)$$

簡明を期すため、入力は一つだけの多項式にしているが、入力はもちろん、多項式の有限集合でもよい。

# 3 ISCZ 法

## 3.1 シンボル付き区間

区間と形式的なシンボルを組み合わせた係数（シンボル付き区間）を導入する。区間は、従来と同様の意味の区間である。シンボルは、アルゴリズム実行中に現れる係数の log（記録）を取るのに使われる。例えば、入力係数が $\frac{1}{3}$ と $\frac{1}{7}$ だったとしよう。これらの精度 3 の区間はそれぞれ $[0.333, 0.334]$ と $[0.142, 0.143]$ である。さて、$\frac{1}{3}$ に対するシンボルを $s$、$\frac{1}{7}$ に対するシンボルを $t$ として、区間と組み合わせると、それぞれ、$[[0.333, 0.334], s]$ と $[[0.142, 0.143], t]$ となる。次に、これらの間の演算、

$$[[0.333, 0.334], s] + [[0.142, 0.143], t] = [[0.333, 0.334] + [0.142, 0.143], s\dot{+}t]$$

と定義する。$[0.333, 0.334] + [0.142, 0.143]$ に対しては通常の区間演算を使う。シンボル部分 $s\dot{+}t$ は再び形式的なシンボルで、加算を実施したことを記録できれば何でもよい。効率的なシンボル付けについては 5.5 節で議論する。

アルゴリズム終了後、最終的なシンボルを正確係数に復元する。先の簡単な例で言えば、もし最終的なシンボルが $s\dot{+}t$ であったとすれば、$s$ に $\frac{1}{3}$ を、$t$ に $\frac{1}{7}$ を代入し、$\dot{+}$ には加算の意味を与えて、$\frac{1}{3} + \frac{1}{7} = \frac{10}{21}$ と復元する。

シンボル付き区間のことを interval-symbol、あるいは単に IS と呼ぶ。

## 3.2 手続き

A を不連続点 0 の代数的アルゴリズムとする。ISCZ 法の手続きは次の通りである。

**R-to-IS** 各入力係数 $r$ を $[[A, B], Symbol_r]$ に変換する。ここに、$[A, B]$ は $r$ の予め定められた精度の区間、$Symbol_r$ は $r$ を表すシンボル (以下、入力シンボルと呼ぶ) である。

**IS 演算** IS 間の演算を次のように実行する:

$$[[A, B], s] + [[C, D], t] = [[A, B] + [C, D], s\dot{+}t]$$

$$[[A, B], s] - [[C, D], t] = [[A, B] - [C, D], s\dot{-}t]$$

$$[[A, B], s] \times [[C, D], t] = [[A, B] \times [C, D], s\dot{\times}t]$$

$$[[A, B], s] \div [[C, D], t] = [[A, B] \div [C, D], s\dot{\div}t]$$

すなわち、区間部分については区間演算を用い、シンボル部分については加算、減算、乗算、除算の形式的なシンボル $\dot{+}, \dot{-}, \dot{\times}, \dot{\div}$ を使って、どういう演算が行なわれたかを記録する。

**正しいゼロ書換え** 任意の IS$[[A,B],s]$ に対し、$A \leq 0 \leq B$ ならば、$s$ をそれに対応する実数 $r(s)$ に復元する。もし、$r(s) = 0$ ならば、次のステップに進む。そうでなければ、精度を上げて **R-to-IS** に戻る。

**IS-to-R** 出力のシンボル部分の中の各入力シンボルにそれぞれ対応する入力係数を代入し、演算シンボルに演算の意味を与えて実数値に復元する。

この手法を ISCZ 法 (IS method with correct zero rewriting) と呼ぶ。

さて、ある精度 $j$ で $Int(f)_j$ を $Stab(\mathcal{A})$ に入力したときの実行過程は、もしアルゴリズム中のすべてのゼロ書換えが正しいならば、真の出力 $f$ を A に入力したときの実行過程と完全に一致する。ISCZ 法では、各ゼロ書換えにおいて、それが正しいかどうかを確認する。さらに、定理 1 により、すべてのゼロ書換えが正しくなる精度が存在する。従って、ISCZ 法は有限ステップで終了し、その出力の各 IS 係数のシンボルは正しい正確係数を与える。

これを定理にまとめる。([4])

**定理 2 (ISCZ 法の停止性と正当性)**
$A$ が入力 $I$ で正常終了するとせよ。このとき、$A$ に対する *ISCZ* 法は、常に有限ステップで終了し、正しい結果、すなわち、$\mathcal{A}(I)$ の出力と同じ結果を与える。

ISCZ 法の利点は、ISZ 法と違い、出力の正当性を確認する必要がないことである。任意の IS$[[A,B],s]$ に対し、$A \leq 0 \leq B$ でない限り、正確計算をスキップすることができ、浮動小数点計算だけで済む。換言すれば、ゼロでない係数についての正確計算を省略することができる。従って、本手法は、$A \leq 0 \leq B$ でない場合が $A \leq 0 \leq B$ である場合よりも多ければ多いほど有効であるということができる。

# 4　本研究の目的

本研究では以下の 2 点を目的とする。

- Maple14 において以下の 3 つのアルゴリズムに ISCZ 法を適用させる
  - ユークリッドの互除法 (2 つの多項式の最大公約式を求める)
  - スツルムアルゴリズム (実根の個数を求める)
  - グラハムアルゴリズム (平面上の点集合の凸包を求める)
- 今まで数式処理システムで実装されてきた安定化手法を、その他の言語でも実装できるようにする

数式処理システム以外での実装が可能となれば、web システムや組み込みソフトなどにも適用でき、応用の幅を広げることができる。そのため、まず数式処理システム以外での実装方法を考え、数式処理システムで実装する。その後、数式処理システム以外の言語でも実装を行うことを目標とした。

# 5　行列を用いたISCZ法のアルゴリズム実装

## 5.1　実装の考え方

ここでは、実装の際の考え方について説明を行う。なお、説明を簡単にするためにシンボルを付加せず、区間演算のみを行うものとする。
今までの計算法では、例えば $\frac{1}{3}x^2 \times \frac{1}{7}x$ を計算する場合は以下の流れで行われていた。

$$\begin{aligned} & \frac{1}{3}x^2 \times \frac{1}{7}x \\ \rightarrow \quad & [0.333, 0.334]x^2 \times [0.142, 0.143]x \\ = \quad & [0.333, 0.334][0.142, 0.143]x^3 \\ = \quad & [0.047286, 0.047762]x^3 \end{aligned}$$

つまり、

- 次数ごとにまとめる計算
- 1つの次数に区間係数が2つ存在したら区間係数を計算

の順番に計算を行い、答えとなる区間係数を出力していた。しかし、これは数式的な書き方・計算の仕方のため、数式処理システム以外では実装が難しいことが問題としてあげられる。
そこで、数式処理システム以外の言語でも実装が容易な新しい計算法を考えた。今までの計算法と同様に、新しい計算法で $\frac{1}{3}x^2 \times \frac{1}{7}x$ を計算する場合の流れを以下に示す。

$$\begin{aligned} & \frac{1}{3}x^2 \times \frac{1}{7}x \\ A \quad = \quad & \frac{1}{3}x^2 \qquad\qquad\qquad B = \frac{1}{7}x \\ & \qquad\quad 0 \qquad 1 \qquad\qquad\qquad\quad 0 \qquad 1 \\ A \quad = \quad & 2 \begin{pmatrix} 0.333 & 0.334 \end{pmatrix} \qquad B = 1 \begin{pmatrix} 0.142 & 0.143 \end{pmatrix} \\ A \times B \quad = \quad & [A_{2,0} \times B_{1,0}, A_{2,1} \times B_{1,1}] \\ = \quad & [0.333 \times 0.142, 0.334 \times 0.143] \\ = \quad & [0.047286, 0.047762] \rightarrow C_{2+1} = C_3 \\ & \qquad\quad 0 \qquad\quad 1 \\ C \quad = \quad & 3 \begin{pmatrix} 0.0472896 & 0.047762 \end{pmatrix} \\ \rightarrow \quad & [0.047286, 0.047762]x^3 \end{aligned}$$

つまり、

- 各次数の係数を行列に保存
  → n 次の項の係数を n 行目に
- 行列の要素を取り出して計算

という流れである。この方法は、多次元配列が実装可能な言語であれば実装は容易である。詳細な区間の保存方法や演算の方法は以降の節で説明する。

## 5.2 区間化

区間化は入力実数 $x$ に対し $a \leq x \leq b$ となる $a, b$ ならばどのような $a, b$ を用いても良い。しかし、$a, b$ が $x$ から離れるほど区間内に 0 が含まれやすくなり、ゼロ書き換え・シンボル判定の回数が増えてしまう。よって、今回は (小数点以下精度桁 +1) 桁目を切り捨て・切り上げしたものを区間の下界・上界とした。この区間を行列に保存する際には、下界を行列の 0 列目に、上界を 1 列目に、元の実数を表すシンボルを 2 列目に保存することにする。

[1 変数多項式 $f(x)$ の精度桁 $n$ 桁での区間化の流れ]

Input:1 変数多項式 $f(x)$, 精度桁 $n$
Output:区間係数を保存する行列 $A$

$i = 0$ から $f(x)$ の次数まで $i$ を増やしながら以下を繰り返す

$$
\begin{aligned}
r &= f(x) \text{ の } x^i \text{の係数} \\
A_{i,0} &= \frac{\lfloor r \times 10^n \rfloor}{10^n} \\
A_{i,1} &= \frac{\lceil r \times 10^n \rceil}{10^n} \\
A_{i,2} &= sn
\end{aligned}
$$

$\lfloor x \rfloor$ は $x$ 以下の最大の整数を、$\lceil x \rceil$ は $x$ 以上の最小の整数を表す。
$sn$(シンボル番号) に関しては 5.5 節で説明する。

[1 変数多項式 $\frac{1}{3}x^2+\frac{1}{7}x+\frac{2}{9}$ の精度桁 3 桁での区間化例]

Input:1 変数多項式 $\frac{1}{3}x^2+\frac{1}{7}x+\frac{2}{9}$, 精度桁 3

$$
\begin{aligned}
r &= f(x)\ の\ x^0 の係数 = \frac{2}{9} \\
A_{0,0} &= \frac{\lfloor \frac{2}{9}\times 10^3 \rfloor}{10^3} = \frac{\lfloor 0.22222\ldots \times 10^3 \rfloor}{10^3} \\
&= \frac{\lfloor 222.22\ldots \rfloor}{10^3} = \frac{222}{10^3} = 0.222 \\
A_{0,1} &= \frac{\lceil \frac{2}{9}\times 10^3 \rceil}{10^3} = 0.223 \\
r &= f(x)\ の\ x^1 の係数 = \frac{1}{7} \\
&\vdots \\
& \qquad\quad 0 \quad\ 1 \quad\ 2 \\
Output : A &= \begin{matrix} 0 \\ 1 \\ 2 \end{matrix}\begin{pmatrix} 0.222 & 0.223 & 0 \\ 0.142 & 0.143 & 1 \\ 0.333 & 0.334 & 2 \end{pmatrix}
\end{aligned}
$$

## 5.3 区間演算

区間演算は区間が保存されている行列の要素を取り出して行う。

[加算・減算の場合]

Input:$A=\frac{1}{3}, B=\frac{1}{7}$ 精度桁 $n$

Output:$C=A+B$

$$
\begin{aligned}
& & 0 \quad\ 1 \quad\ 2 & \qquad & 0 \quad\ 1 \quad\ 2 \\
A &= 0\ \begin{pmatrix} 0.333 & 0.334 & 1 \end{pmatrix} & B = 0\ \begin{pmatrix} 0.142 & 0.143 & 2 \end{pmatrix} \\
i &= 計算する次数 = 0 \\
C_{i,0} = C_{0,0} &= A_{0,0}+B_{0,0} = 0.333+0.142 = 0.475 \\
C_{i,1} = C_{0,1} &= A_{0,1}+B_{0,1} = 0.334+0.143 = 0.477 \\
C_{i,2} = C_{0,2} &= sn = 3 \\
& & 0 \quad\ 1 \quad\ 2 \\
C &= 0\ \begin{pmatrix} 0.475 & 0.477 & 3 \end{pmatrix}
\end{aligned}
$$

$sn$(シンボル番号) に関しては 5.5 節で説明する。

減算も同様なので省略する。

[乗算の場合]

Input:$A = \frac{1}{3}x^2, B = \frac{1}{7}x$ 精度桁 $n$
Output:$C = A \times B$

$$
\begin{array}{rcl}
 & & \begin{array}{ccc} 0 & 1 & 2 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 2 \end{array} \\
A & = & 2 \begin{pmatrix} 0.333 & 0.334 & 1 \end{pmatrix} \quad B = 1 \begin{pmatrix} 0.142 & 0.143 & 2 \end{pmatrix} \\
i & = & \text{計算する } A \text{ の次数} = 2 \\
j & = & \text{計算する } B \text{ の次数} = 1 \\
C_{i+j,0} = C_{3,0} & = & A_{2,0} \times B_{1,0} = 0.333 \times 0.142 = 0.047286 \\
C_{i+j,1} = C_{3,1} & = & A_{2,1} \times B_{1,1} = 0.334 \times 0.143 = 0.047762 \\
C_{i+j,2} = C_{3,2} & = & sn = 3 \\
 & & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\
C & = & 3 \begin{pmatrix} 0.047286 & 0.047762 & 3 \end{pmatrix}
\end{array}
$$

$sn$(シンボル番号) に関しては 5.5 節で説明する。

[除算の場合]

Input:$A = \frac{1}{3}x^2, B = \frac{1}{7}x$ 精度桁 $n$
Output:$C = \frac{A}{B}$

$$
\begin{array}{rcl}
 & & \begin{array}{ccc} 0 & 1 & 2 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 2 \end{array} \\
A & = & 2 \begin{pmatrix} 0.333 & 0.334 & 1 \end{pmatrix} \quad B = 1 \begin{pmatrix} 0.142 & 0.143 & 2 \end{pmatrix} \\
i & = & \text{計算する } A \text{ の次数} = 2 \\
j & = & \text{計算する } B \text{ の次数} = 1 \\
C_{i-j,0} = C_{1,0} & = & \dfrac{A_{2,0}}{B_{1,1}} = \dfrac{0.333}{0.143} = 2.328671 \\
C_{i-j,1} = C_{1,1} & = & \dfrac{A_{2,1}}{B_{1,0}} = \dfrac{0.334}{0.142} = 2.352112 \\
C_{i-j,2} = C_{1,2} & = & sn = 3 \\
 & & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\
C & = & 1 \begin{pmatrix} 2.328671 & 2.352112 & 3 \end{pmatrix}
\end{array}
$$

$sn$(シンボル番号) に関しては 5.5 節で説明する。

## 5.4　ゼロ書き換え

ゼロ書き換えを行う条件は「区間の下界と上界の間に 0 を含むこと」であるが、実装では「区間の下界と上界の積が 0 以下であること」とする。

$$X = [[a, b], s] \text{ のとき}$$
$$(a \leq 0)\ and\ (0 \leq b) \Leftrightarrow ab \leq\ 0$$

このとき、$ab \leq 0$ を満たすならば、シンボル $s$ を用いて本当に 0 であるかの評価を行う。

## 5.5　シンボルリスト

3.1 節で述べたようなシンボルの保存方法を行うと、複雑な計算を行った際に IS のシンボル部分が膨張してしまうことがある。それを防ぐため、IS に用いるシンボルは全て整数とし、計算の経過を行番号に対応させて保存する行列 (シンボルリスト) を用いることとする。シンボルリストについては ([5],[6]) において提案されている。

シンボルリストには $n$ 行 3 列の行列を用い、0 列目・1 列目には数値もしくはシンボル、2 列目には演算を保存する。もし、シンボルが表すものが実数であれば、0 列目に実数、1 列目と 2 列目に $-1$ を保存し、演算ではないことを示す。シンボル番号は、IS 演算を行うごとに新しい番号をつけていく。

[数式処理システムでのシンボルリスト保存]

- シンボル番号 $s$ が入力実数 $A$ を表すとき

$$\begin{array}{c} \quad 0 \quad 1 \quad 2 \\ s\ \begin{pmatrix} A & -1 & -1 \end{pmatrix} \end{array}$$

※ $-1$ は空要素を表す

- シンボル番号 $s$ が $t \star u$ の演算を表すとき
  ($t, u$ はシンボル番号、$\star$ は $+, -, \times, \div$ のいずれか)

$$\begin{array}{c} \quad 0 \quad 1 \quad 2 \\ s\ \begin{pmatrix} t & u & \star \end{pmatrix} \end{array}$$

[数式処理システムでのシンボルリスト保存の例]

$$\begin{aligned}\frac{1}{3}+\frac{1}{7}\times\frac{2}{9} &= [[0.333,0.334],0]+[[0.142,0.143],1]\times[[0.222,0.223],2]\\ &= [[0.333,0.334],0]+[[0.031524,0.031889],3]\\ &= [[0.364524,0.365889],4]\end{aligned}$$

の計算経過をシンボルリストに保存する場合、

$$SL=\begin{array}{c} \\ 0\\1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{ccc}0 & 1 & 2\end{array}\\ \begin{pmatrix}\frac{1}{3} & -1 & -1\\ \frac{1}{7} & -1 & -1\\ \frac{2}{9} & -1 & -1\\ 1 & 2 & \times\\ 0 & 3 & +\end{pmatrix}\end{array}$$

しかし、入力実数が分数 (有理数) の場合、数式処理システム以外では分数を分数のまま保存するのは難しい。そのため、0 行目に分子、1 行目に分母を保存するように変更した。$\sqrt{2}$ や $\pi$ などの無理数については、評価の際の計算が正確に行えないため、今回は考えないこととした。

[数式処理システム以外の言語でのシンボルリスト保存]

- シンボル番号 $s$ が入力実数 $\frac{A}{B}$ を表すとき

$$\begin{array}{c}\\ s\end{array}\begin{array}{c}\begin{array}{ccc}0 & 1 & 2\end{array}\\ \begin{pmatrix}A & B & -1\end{pmatrix}\end{array}$$

※ $-1$ は空要素を表す

- シンボル番号 $s$ が $t\star u$ の演算を表すときは数式処理システムと同じ

[数式処理システム以外の言語でのシンボルリスト保存の例]

$$\begin{aligned}\frac{1}{3}+\frac{1}{7}\times\frac{2}{9} &= [[0.333,0.334],0]+[[0.142,0.143],1]\times[[0.222,0.223],2]\\ &= [[0.333,0.334],0]+[[0.031524,0.031889],3]\\ &= [[0.364524,0.365889],4]\end{aligned}$$

の計算経過をシンボルリストに保存する場合、

$$SL=\begin{array}{c} \\ 0\\1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{ccc}0&1&2\end{array}\\ \begin{pmatrix}1&3&-1\\1&7&-1\\2&9&-1\\1&2&\times\\0&3&+\end{pmatrix}\end{array}$$

## 5.6 シンボルリストの評価方法

### 5.6.1 行列法

5.5 節で保存したシンボルリストを用いて、シンボル番号 4 を評価する手順によって説明する。

$$SL=\begin{array}{c} \\ 0\\1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{ccc}0&1&2\end{array}\\ \begin{pmatrix}\frac{1}{3}&-1&-1\\\frac{1}{7}&-1&-1\\\frac{2}{9}&-1&-1\\1&2&\times\\0&3&+\end{pmatrix}\end{array}$$

流れとしては、シンボルリストから評価行列を作り、その評価行列を用いて計算することになる。

各行の 0 列目にはその行にいくつのシンボルリストから取った 3 つ組が存在するかを保存する。例えば 0 行目には評価を行うシンボル番号 4 に対応するシンボルリストの行のみを入れるため、3 つ組の数は 1 となり、0 行 0 列目に 1 が保存される。

$$eva=\begin{array}{c} \\ 0\end{array}\begin{array}{c}\begin{array}{cccc}0&1&2&3\end{array}\\ \begin{pmatrix}1&0&3&+\end{pmatrix}\end{array}$$

eva 行列の $n$ 行 $3m+3$ 列目 $(m\geq 0, m\in\mathbb{Z})$ を読み、演算記号ならばシンボルリストから $3m$ 列目・$3m+1$ 列目のシンボルに対応するシンボルリストの行を $n+1$ 行目に左詰めで入れる。演算記号ではなく $-1$ ならば何も行わ

ずスキップする。$n$ 行目の全ての列が終わったら $n+1$ 行目以降にも同様の操作を行い、$n$ 行目に演算シンボルがなくなるまで繰り返す。

$$eva = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 3 & + & & & \\ 2 & \frac{1}{3} & -1 & -1 & 1 & 2 & \times \end{pmatrix} \end{array}$$

$$eva = \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 3 & + & & & \\ 2 & \frac{1}{3} & -1 & -1 & 1 & 2 & \times \\ 2 & \frac{1}{7} & -1 & -1 & \frac{2}{9} & -1 & -1 \end{pmatrix} \end{array}$$

評価行列が完成してから演算を行う。$n$ 行の評価行列での演算は $n$ 行目,$n-1$ 行目,$\cdots$,0 行目の順に行い、演算シンボルがあれば 1 行下に保存されている実数を用いて計算を行うこととする。計算結果は演算シンボルが属する 3 つ組に上書きする。

$$eva = \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 3 & + & & & \\ 2 & \frac{1}{3} & -1 & -1 & 1 & 2 & \times \\ 2 & \frac{1}{7} & -1 & -1 & \frac{2}{9} & -1 & -1 \end{pmatrix} \end{array}$$

$$\downarrow$$

$$eva = \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 3 & + & & & \\ 2 & \frac{1}{3} & -1 & -1 & \frac{2}{63} & -1 & -1 \\ 2 & \frac{1}{7} & -1 & -1 & \frac{2}{9} & -1 & -1 \end{pmatrix} \end{array}$$

また、同じシンボルを複数回評価するのは無駄であるため、一度評価されたシンボルはシンボルリストに実数として上書きする。

$$SL = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{pmatrix} \frac{1}{3} & -1 & -1 \\ \frac{1}{7} & -1 & -1 \\ \frac{2}{9} & -1 & -1 \\ 1 & 2 & \times \\ 0 & 3 & + \end{pmatrix} \end{array} \rightarrow SL = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{pmatrix} \frac{1}{3} & -1 & -1 \\ \frac{1}{7} & -1 & -1 \\ \frac{2}{9} & -1 & -1 \\ \frac{2}{63} & -1 & -1 \\ 0 & 3 & + \end{pmatrix} \end{array}$$

この操作を繰り返し、0 行目が計算された時の 0 行 1 列目が評価結果となる。

### 5.6.2 式列法

式列法の基本形は次のようになる。

$$eva = \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} \star & s & \text{“}s\text{”} & t & \text{“}s\text{”} & u \end{pmatrix} \end{array}$$

ここで、$s,t,u$ はシンボル番号を、“$s$” は次の数がシンボルであることを、$\star$ は演算記号を表す。この行列を 0 列目から走査し、以下の条件に当てはまったら操作を加える。

- $n$ 列目が “$s$” かつ $n+1$ 列目のシンボルが演算を表すとき
  $n$ 列目を起点として基本形に展開する。$n+2$ 列目以降を右に 4 つシフトする。

$$
\begin{array}{rc}
 & \begin{array}{ccccccc} \cdots & n-2 & n-1 & n & n+1 & n+2 & n+3 \quad \cdots \end{array} \\
eva = & \begin{pmatrix} \cdots & \star & s & \text{“}s\text{”} & t & \text{“}s\text{”} & u \quad \cdots \end{pmatrix}
\end{array}
$$

$$\downarrow$$

$$
\begin{array}{rc}
 & \begin{array}{ccccccccccc} \cdots & n-2 & n-1 & n & n+1 & n+2 & n+3 & n+4 & n+5 & n+6 & \cdots \end{array} \\
eva = & \begin{pmatrix} \cdots & \star & s & \star & t & \text{“}s\text{”} & v & \text{“}s\text{”} & w & \text{“}s\text{”} & \cdots \end{pmatrix}
\end{array}
$$

- $n$ 列目が “$s$” かつ $n+1$ 列目のシンボルが実数を表すとき
  シンボルを表す “$s$” を実数を表す “$r$” に変え、後ろに実数を入れる。

$$
\begin{array}{rc}
 & \begin{array}{ccccccc} \cdots & n & n+1 & n+2 & n+3 & n+4 & n+5 \quad \cdots \end{array} \\
eva = & \begin{pmatrix} \cdots & \star & s & \text{“}s\text{”} & t & \text{“}s\text{”} & u \quad \cdots \end{pmatrix}
\end{array}
$$

$$\downarrow$$

$$
\begin{array}{rc}
 & \begin{array}{ccccccc} \cdots & n & n+1 & n+2 & n+3 & n+4 & n+5 \quad \cdots \end{array} \\
eva = & \begin{pmatrix} \cdots & \star & s & \text{“}r\text{”} & p & \text{“}s\text{”} & u \quad \cdots \end{pmatrix}
\end{array}
$$

- $n+2$ 列目と $n+4$ 列目が共に “$r$” のとき
  演算を行い、$n+6$ 列目以降を左に 4 つシフトする。

$$
\begin{array}{rc}
 & \begin{array}{ccccccccc} \cdots & n & n+1 & n+2 & n+3 & n+4 & n+5 & n+6 & n+7 \quad \cdots \end{array} \\
eva = & \begin{pmatrix} \cdots & \star & s & \text{“}r\text{”} & p & \text{“}r\text{”} & q & \text{“}s\text{”} & u \quad \cdots \end{pmatrix}
\end{array}
$$

$$\downarrow$$

$$
\begin{array}{rc}
 & \begin{array}{ccccc} \cdots & n & n+1 & n+2 & n+3 \quad \cdots \end{array} \\
eva = & \begin{pmatrix} \cdots & \text{“}r\text{”} & r & \text{“}s\text{”} & u \quad \cdots \end{pmatrix}
\end{array}
$$

  また、演算後は行列法と同様にシンボルリストにも演算結果を上書きする。

この操作を繰り返し、0 列目が “$r$” になった時の 1 列目が評価結果となる。

# 6　実験

この実験で使用する PC および実行環境は以下の通り。

使用コンピュータ
OS:Windows 7 Professional
CPU:Intel(R) Pentium(R) CPU G840 @ 2.80GHz
メモリ:4.00GB
使用ソフト:数式処理ソフト Maple14
JavaScript を動作させるブラウザ:Chrome38.0.2125.104m

## 6.1　ユークリッドの互除法への適用

### 6.1.1　ユークリッドの互除法

ユークリッドの互除法とは、2 つの自然数や整式の最大公約数を求めるアルゴリズムである。2 つの自然数や整式を余りが 0 になるまで除算し、余りが 0 になった直前の割る数が求める最大公約数となる。

[ユークリッドの互除法の流れ]

Input: 自然数 $a, b(a \geq b)$
Output: $a$ と $b$ の最大公約数

Step1/ $\frac{a}{b}$ の余り $r$ を求める

$$\begin{aligned} q &= \lfloor \frac{a}{b} \rfloor \\ r &= a - b \times q \end{aligned}$$

($\lfloor x \rfloor$ は $x$ 以下の最大の整数を表す)
Step2/ $b$ を $a$、$r$ を $b$ に置き換える
Step3/ $b = 0$ であれば、Step4 へ進む。そうでなければ Step1 へもどる
Step4/ $a$ が最大公約数である

[ユークリッドの互除法の例]

Input:$25, 10$
大きい方の 25 を $a$、小さい方の 10 を $b$ とする。
$\lfloor \frac{25}{10} \rfloor = 2$ なので、$q = 2$ とする。
$r = 25 - 10 \times 2 = 5$
$b$ を $a$、$r$ を $b$ に置き換えるので、$a = 10, b = 5$ となる。
$b = 0$ でないのでもう 1 度繰り返す。
$\lfloor \frac{10}{5} \rfloor = 2$ なので、$q = 2$ とする。
$r = 10 - 5 \times 2 = 0$
$b$ を $a$、$r$ を $b$ に置き換えるので、$a = 5, b = 0$ となる。
$b = 0$ であるので、25 と 10 の最大公約数は 5 であると求められた。
Output:5

### 6.1.2 実験方法

- 1 変数多項式のユークリッドの互除法を行う。
- 以下の 4 つの方法で計算開始から出力までの時間・メモリ使用量を計測する。ISCZ 法については評価行列の最大要素数も計測する。
  1. ISCZ 法のユークリッドの互除法アルゴリズムに、行列法の評価方法を組み込んだもの (M 行列)
  2. ISCZ 法のユークリッドの互除法アルゴリズムに、式列法の評価方法を組み込んだもの (M 式列)
  3. 自作の正確演算でのユークリッドの互除法アルゴリズム (M 厳密)
  4. Maple に組み込まれている GCD 関数 (M 関数)
- 入力する多項式は次数のみ指定し、係数はランダムとして生成した
- ISCZ 法の開始精度は 1 桁とし、0 判定での評価の結果「本当は 0 でない」と判断した場合は、精度桁を 1 桁ずつ増やす

表 1: 実験結果 (安定化計算・有理係数)

| 入出力 次数 | SP | ZR | SL | M 行列 | | | M 式列 | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Time | Memory | Elements | Time | Memory | Elements |
| 5-3-2 | 3 | 7 | 51 | 0 | 208.51k | 102 | 0 | 139.71k | 42 |
| 9-7-3 | 8 | 51 | 165 | 47 | 2.67M | 204 | 31 | 2.92M | 106 |
| 10-8-6 | 4 | 25 | 141 | 15 | 0.90M | 178 | 31 | 1.03M | 82 |
| 39-26-19 | 9 | 341 | 1495 | 436 | 30.83M | 4778 | 453 | 28.98M | 234 |

表 2: 実験結果 (厳密計算・有理係数)

| 入出力 次数 | SP | ZR | SL | M 厳密 | | M 関数 | |
|---|---|---|---|---|---|---|---|
| | | | | Time | Memory | Time | Memory |
| 5-3-2 | 3 | 7 | 51 | 0 | 8.43k | 0 | 10.94k |
| 9-7-3 | 8 | 51 | 165 | 0 | 35.31k | 0 | 12.09k |
| 10-8-6 | 4 | 25 | 141 | 0 | 18.46k | 0 | 12.52k |
| 39-26-19 | 9 | 341 | 1495 | 0 | 164.66k | 0 | 28.18k |

### 6.1.3 実験結果

実験結果表中の略号は以下の通り。

入出力次数:(入力した多項式 A の次数)
-(入力した多項式 B の次数)-(出力された多項式の次数)
SP:出力精度
(開始精度は有理係数では 3 桁、無理係数では 1 桁)
ZR:ゼロ書き換え回数　SL:シンボルリストの長さ
Time:ミリ秒　Memory:k=kib M=Mib　Elements:評価行列の要素数

表 1 と表 2 は入力多項式の係数が有理数だけのものである。
有理係数での実験では行列法と式列法についてはどちらが優位とは決められなかった。
また、自作の正確演算アルゴリズム、Maple に組み込まれている GCD 関数と比べると時間・メモリ使用量共に ISCZ 法を用いない方が優位であった。

表 3 と表 4 は入力多項式の係数に無理数が含まれるものである。
無理係数での実験では行列法よりも式列法の方が優位であった。
しかし、やはり自作の正確演算アルゴリズム、Maple に組み込まれている GCD 関数と比べると、時間・メモリ使用量共に ISCZ 法を用いない方が優位

表 3: 実験結果 (安定化計算・無理係数)

| 入出力 次数 | SP | ZR | SL | M 行列 | | | M 式列 | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Time | Memory | Elements | Time | Memory | Elements |
| 3-2-1 | 2 | 9 | 31 | 312 | 15.88M | 77 | 94 | 9.84M | 42 |
| 7-4-3 | 3 | 21 | 75 | 265 | 13.37M | 76 | 203 | 9.70M | 42 |
| 11-6-3 | 5 | 41 | 175 | 1607 | 86.76M | 178 | 1216 | 69.57M | 82 |
| 15-11-5 | 11 | 133 | 369 | 8658 | 472.43M | 306 | 5584 | 297.22M | 146 |

表 4: 実験結果 (厳密計算・無理係数)

| 入出力 次数 | SP | ZR | SL | M 厳密 | | M 関数 | |
|---|---|---|---|---|---|---|---|
| | | | | Time | Memory | Time | Memory |
| 3-2-1 | 2 | 9 | 31 | 31 | 1.41M | 0 | 1.68M |
| 7-4-3 | 3 | 21 | 75 | 63 | 1.29M | 47 | 1.46M |
| 11-6-3 | 5 | 41 | 175 | 0 | 0.76M | 62 | 4.48M |
| 15-11-5 | 11 | 133 | 369 | 46 | 4.76M | 390 | 25.13M |

であった。
このことから、無理係数に限っては式列法が有効であると考えられる。また、
ISCZ 法自体がユークリッドの互除法には効果が薄いということもわかった。

## 6.2　スツルムアルゴリズムへの適用

### 6.2.1　スツルムアルゴリズム

スツルムアルゴリズム ([7]) とは、1 変数多項式の実解の個数を計算するアルゴリズムである。入力多項式を $f(x)$ としたとき、以下の操作を行うことで、$f(x)$ の実解の個数を求めることができる。

[スツルムアルゴリズムの流れ]

Input:1 変数多項式 $f(x) = 0$, 区間 [a,b]
Output:$f(x)$ の区間 [a,b] 内に存在する実根の個数

Step1/スツルム列を生成する

$$\begin{aligned} f_0(x) &= f(x) \\ f_1(x) &= f'(x) \\ f_{n+2}(x) &= -\{f_n(x) - f_{n+1}(x)g_{n+1}(x)\} \quad (n \geq 0) \\ &\vdots \\ f_m(x) &= -\{f_{m-2}(x) - f_{m-1}(x)g_{m-1}(x)\} \\ f_{m+1}(x) &= -\{f_{m-1}(x) - f_m(x)g_m(x)\} = 0 \end{aligned}$$

つまり、$f_n$ を $f_{n+1}$ で割った余りの符号を変えたものを $f_{n+2}$(x) とし、余りが 0 となるまで繰り返す。(これは有限回で停止する)
このときの $f_0(x), f_1(x), \ldots, f_m(x)$ をスツルム列と呼ぶ。

Step2/符号変化数の計算方法
生成したスツルム列

$$f_0(x), f_1(x), \ldots, f_m(x)$$

の x に t を代入した際の値を、$f_0$ から順に見ていったときの符号変化数を $N(t)$ とする。

Step3/区間 [a,b] 内の実根数を計算する
$f(x) = 0$ の区間 [a,b] 内の実根数は

$$|N(a) - N(b)|$$

で求められる。

[スツルムアルゴリズムの例]

例:$f(x) = 2x^3 + 3x^2 + 5x + 7$、区間 $[-2, 2]$ の場合
Input:1 変数多項式 $2x^3 + 3x^2 + 5x + 7 = 0$, 区間 [-2, 2]
～スツルム列～

$$
\begin{aligned}
f_0(x) &= 2x^3 + 3x^2 + 5x + 7 \\
f_1(x) &= 6x^2 + 6x + 5 \\
f_2(x) &= -\frac{7}{3}x - \frac{37}{6} \\
f_3(x) &= -\frac{3043}{98} \\
f_4(x) &= 0
\end{aligned}
$$

～符号変化数～

| | $x = -2$ | $x = 2$ |
|---|---|---|
| $f_0(x)$ | $-7$ | 45 |
| $f_1(x)$ | 17 | 41 |
| $f_2(x)$ | $-\frac{3}{2}$ | $-\frac{65}{6}$ |
| $f_3(x)$ | $-\frac{3043}{98}$ | $-\frac{3043}{98}$ |
| 変化数 | 2 | 1 |

よって、$f(x) = 2x^3 + 3x^2 + 5x + 7$ の区間 $[-2, 2]$ にある実根の個数は 1 個である。
Output:1 個

### 6.2.2 実験方法

- 1 変数多項式に対してスツルムアルゴリズムを適用し、その実解の個数を計算する。
- 以下の 3 つの方法で計算開始から出力までの時間・メモリ使用量を計測する。ISCZ 法については評価行列の最大要素数も計測する。
  1. ISCZ 法のスツルムアルゴリズムに、行列法の評価方法を組み込んだもの (M 行列)
  2. ISCZ 法のスツルムアルゴリズムに、式列法の評価方法を組み込んだもの (M 式列)

表 5: スツルム実験結果

| 入力次数 | 出力精度 | ゼロ判定数 | シンボルリスト長 | 実行時間 (秒) | | | |
|---|---|---|---|---|---|---|---|
| | | | | M 行列 | M 式列 | M 厳密 | JS |
| 4 | 3(4) | 23 | 45 | 0.031 | 0.015 | 0.016 | 0.016 |
| 5 | 4(5) | 48 | 86 | 0.063 | 0.031 | 0 | 0.016 |
| 7 | 5 | 86 | 144 | 0.093 | 0.078 | 0.015 | 0.016 |
| 10 | 7 | 280 | 288 | 0.328 | 0.296 | 0.016 | **** |

- 4・5 次式入力では Maple と JS で出力精度が異なったため、
  JS での精度を () 内に記載
- 10 次式においては JS で出力が得られなかった
  詳細は 6.2.3 項に記載

3. 自作の正確演算でのスツルムアルゴリズム (M 厳密)
4. JavaScript で実装した ISCZ スツルムアルゴリズムに、式列法の評価方法を組み込んだもの (JS)

- 入力する多項式は実解の個数を指定し、その実解および重複度はランダムとして生成した
- ISCZ 法の開始精度は 2 桁とし、0 判定での評価の結果「本当は 0 でない」と判断した場合は、精度桁を 1 桁ずつ増やす

スツルムアルゴリズムでは数式処理システム以外の言語として JavaScript を用いて実験を行った。JavaScript を選択した理由は、

- web ブラウザさえあれば PC の OS によらず実行が可能である
- 動的配列の宣言が容易である

からである。

2 番目の理由にある「動的配列」とは変数宣言時に配列の要素数を定義しなくてもよい配列のことである。今回の ISCZ 法では、必要なシンボルリストの行数が不明となっている。そのため、変数宣言時に要素数を定義しなければならない「静的配列」で実装する場合、シンボルリスト行列の宣言時に巨大な領域を持つように宣言することとなる。その場合、メモリ使用量が増えてしまうため、動的配列の宣言が容易な言語を用いるのがよいと考えた。

### 6.2.3 実験結果

実験結果を表 5 に記載した。Maple と JavaScript で出力精度が異なっているが、今回は正しい出力をしているか・どの程度の時間がかかったかに重点を置いたため無視した。

Maple の安定化演算では厳密計算のほうが短時間で計算できたが、JavaScript で実装した安定化演算は 7 次式までにおいて Maple の厳密計算とほぼ同じ時間で計算を終えた。

JavaScript においては、小数は小数点以下 15 桁で切り捨てる仕様となっている。そのため、各次数の係数が小さく ($10^{-15}$ 程度以下で) 区間化が不可能な場合や、0 判定での評価の結果において「0 でない」が続き、精度桁が 15 桁以上となった場合には、正確な出力を行うことが不可能となる。

## 6.3 グラハムアルゴリズムへの適用

### 6.3.1 グラハムアルゴリズム

グラハムアルゴリズム ([8]) とは、平面上の点 $(x, y)$ の集合 $S$ を入力した際に、その凸包 (全ての点を内部に含む最小の凸多角形) を求めるアルゴリズムである。

[グラハムアルゴリズムの流れ]

Input:平面上の点 $(x, y)$ の集合 $S$(点の数は $n$ 個)
Output:$S$ の凸包 $C$

Step1/基準点を決定する
集合 $S$ の中で $y$ 座標が最小の点を基準点 $(P_0)$ とする。

Step2/偏角によって点をソートする
基準点 $(P_0)$ からその他の各点の偏角を求め、
偏角が小さい方から順に $P_1, P_2, \ldots, P_{n-1}$ とする。
点 $P_0$ から見た点 $Q$ の偏角 $A$ は、

$$A = \frac{Q_x - P_{0x}}{Q_y - P_{0y}}$$

で求める。

Step3/凸包を構成する点を検索する
$3 \leq i \leq n-1$ に対して、点 $P_{i-1}$ と線分 $P_{i-2}P_i$ の位置関係を考える。そのために、

$$R = (P_{i-1_x} - P_{i-2_x}) \times (P_{i_y} - P_{i-2_y}) - (P_{i-1_y} - P_{i-2_y}) \times (P_{i_x} - P_{i-2_x})$$

を計算し、
$R \geq 0$(点 $P_{i-1}$ が線分 $P_{i-2}P_i$ の右側にある状態) ならば、そのまま残し次の $i$ に行く.。そうでない (左側にある状態) ならば、点 $P_{i-1}$ を削除し、$P_i$ 以降を詰める。
最終的に残った点が $S$ の凸包 $C$ を構成する点となる

[凸包の例]

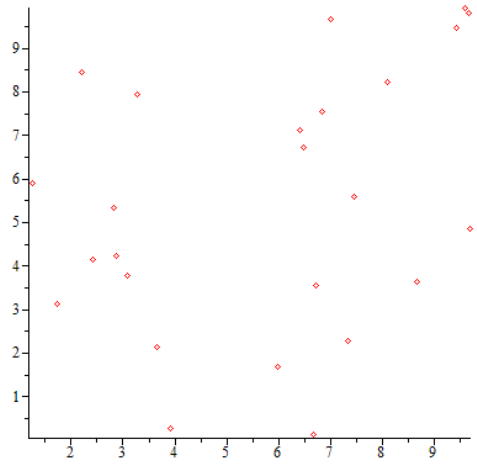

図 1: 点集合 $S$(Input)

⇓

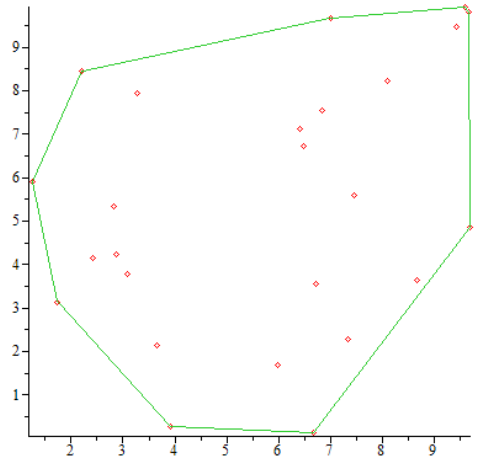

図 2: 点集合 $S$ の凸包 $C$(Output)

### 6.3.2 実験方法

- ランダムに生成した点 $(x, y)$ の集合の凸包を
  グラハムアルゴリズムを用いて計算する
- 以下のそれぞれの方法について、計算開始から出力までの時間を計測

する

1. ISCZ 法のグラハムアルゴリズムに、行列法の評価方法を組み込んだもの (M 行列)
2. ISCZ 法のグラハムアルゴリズムに、式列法の評価方法を組み込んだもの (M 式列)
3. 自作の正確演算でのグラハムアルゴリズム (M 厳密)

- 入力する点集合は以下のそれぞれの中から指定した点数をランダムに生成する
  1. 有 A/$x, y : \dfrac{1 \sim 1000}{3 \sim 100}$ $\quad x^2 + y^2 < 100$
  2. 有 B/$x, y : \dfrac{1 \sim 1000}{3 \sim 100}$ $\quad x^2 + y^2 < 100 \quad y > \dfrac{x}{2} \quad y < 2x$
  3. 無 C/$x, y : \sqrt{\dfrac{1 \sim 10000}{3 \sim 100}}$ $\quad x^2 + y^2 < 100$
  4. 無 D/$x, y : \sqrt{\dfrac{1 \sim 10000}{3 \sim 100}}$ $\quad x^2 + y^2 < 100 \quad y > \dfrac{x}{2} \quad y < 2x$

  入力を扇形にすることで、凸包が半径の直線に近づき、低い精度において、グラハムアルゴリズム Step3 の点を残すか・削除するかの判定で誤判定を生じやすくなる。

- ISCZ 法の開始精度は 2 桁とし、0 判定での評価の結果「本当は 0 でない」と判断した場合は、精度桁を 1 桁ずつ増やす

#### 6.3.3 実験結果

有理数点での計算 (表 6) は、厳密計算の方が ISCZ 法より高速に計算することができた。表に載せた点数以上も実験を行おうとしたが、Object Too Large エラーが発生し、計算を終えることができなかった。これは点数が多くなることで計算回数が多くなり、シンボルリスト行列が膨張したことが原因であると考えられる。

一方、無理数点での計算 (表 7) においては、ISCZ 法が厳密計算と比べ高速に計算できることが確認できた。こちらについては厳密計算の計算時間の関係で 500 点でやめてしまったが、計算時間の増え方から、点数を増やすほど有効性が増すと考えられる。また、今回使用した無理数は $\sqrt{n}$ のみであるので、今後は $e$ や $\pi$ などを含めた実験も行う。

表 6: グラハム実験結果 (有理数点)

| 実験条件 | 入力点数 | 出力精度 | ゼロ判定数 | シンボルリスト長 | 実行時間 (秒) | | |
|---|---|---|---|---|---|---|---|
| | | | | | M 行列 | M 式列 | M 厳密 |
| 有 A | 250 | 5 | 3 | 128382 | 3.400 | 3.542 | 0.281 |
| 有 A | 500 | 6 | 4 | 506840 | 23.743 | 24.305 | 1.623 |
| 有 A | 750 | 6 | 4 | 1135340 | 51.418 | 52.775 | 2.730 |
| 有 A | 1000 | 7 | 5 | 2013854 | 98.514 | 99.591 | 6.334 |
| 有 B | 250 | 6 | 4 | 128368 | 5.663 | 5.803 | 0.281 |
| 有 B | 500 | 7 | 5 | 506882 | 37.752 | 37.518 | 1.747 |
| 有 B | 750 | 6 | 4 | 1135361 | 31.450 | 31.543 | 2.855 |
| 有 B | 1000 | 7 | 5 | 2013861 | 97.142 | 97.641 | 6.786 |
| 有 B | 1125 | 7 | 5 | 2546854 | 78.313 | 78.530 | 6.225 |

表 7: グラハム実験結果 (無理数点)

| 実験条件 | 入力点数 | 出力精度 | ゼロ判定数 | シンボルリスト長 | 実行時間 (秒) | | |
|---|---|---|---|---|---|---|---|
| | | | | | M 行列 | M 式列 | M 厳密 |
| 無 C | 100 | 5 | 3 | 21310 | 1.669 | 1.388 | 232.832 |
| 無 C | 200 | 5 | 3 | 82682 | 5.616 | 5.990 | 900.953 |
| 無 C | 300 | 5 | 3 | 184068 | 9.141 | 9.189 | 1903.212 |
| 無 C | 500 | 6 | 4 | 506861 | 22.683 | 22.791 | 7913.276 |
| 無 D | 100 | 4 | 2 | 21310 | 1.498 | 1.466 | 237.091 |
| 無 D | 200 | 5 | 3 | 82689 | 6.147 | 5.413 | 1138.683 |
| 無 D | 300 | 5 | 3 | 184075 | 12.121 | 12.059 | 3124.903 |
| 無 D | 500 | 6 | 4 | 506854 | 16.068 | 16.240 | 7876.396 |

# 7　まとめと考察

今回の研究では、ユークリッドの互除法、スツルムアルゴリズム、グラハムアルゴリズムに ISCZ 法を組み込み、同手法の有効性を検証した。その結果、ユークリッドの互除法およびスツルムアルゴリズムについては、厳密計算の方が高速に計算できることがわかった。また、グラハムアルゴリズムについては、有理数点では厳密計算の方が高速に計算できたが、無理数点では ISCZ 法の方が高速に計算できた。しかし、シンボルリストの膨張により計算出来なくなることがあったため、シンボルの保存方法についても新たな方法を考えたい。

また、数式処理システムのみで実装されてきた ISCZ 法を用いたアルゴリズムを、数式処理システムではない JavaScript で実装することに成功した。しかし、「実験可能なのは有理数のみである」ことや、「小数点以下 15 桁しか保存できない」という制限があるため、この制限をなくす、もしくは広げる工夫をすることが求められる。さらに、C 言語などの他言語でも実装し実験を行いたい。

# 謝辞

今回の研究を行うにあたりアドバイスを下さった白柳潔教授、並木誠准教授、足立智子准教授、東京理科大学 関川浩教授および白柳研究室の皆様に感謝いたします。

# 参考文献


[1] 白柳 潔: 『コンピュータのカオスをおさえる-新しい「安定」計算術』(コミュニケーションサイエンスシリーズ 1), NTT コミュニケーション科学基礎研究所監修, NTT 出版株式会社, 東京, 2003

[2] 白柳 潔, 関川 浩: 安定化理論における台収束の応用について, 京都大学数理解析研究所講究録 **1568**, 2007, 20–26

[3] 白柳 潔: 代数的アルゴリズムの安定化理論, コンピュータソフトウェア, **Vol.19 No.3**, 2002, 49–65

[4] 白柳 潔, 関川 浩: 安定化理論に基づく ISCZ 法の凸包構成への応用, 京都大学数理解析研究所講究録 **1815**, 2012, 133–142

[5] 白柳 潔, 関川 浩: 安定化理論に基づく ISCZ 法の有効性について, 京都大学数理解析研究所講究録 **1814**, 2012, 29–35

[6] 白柳 潔, 関川 浩: 安定化理論に基づく log method について, 京都大学数理解析研究所講究録 **1666**, 2009, 98–105

[7] 高木 貞治:「第 3 章 スツルムの問題, 根の計算」, 『代数学講義 改訂新版』, 共立出版, 1965, 81–119

[8] R.L.Graham: An efficient algorithm for determining the convex hull of a finite planar set. , Information Processing Letters, **1(4)**, 1972, 132–133

# 付録 1:Maple プログラムソース

## ユークリッドの互除法 (厳密計算)

| 入力引数 | 内容 |
|---|---|
| a | 多項式 A |
| b | 多項式 B |

```
euclid := proc (a, b)
    local p, q, r;

    p := a;
    q := b;

    while q <> 0 do
        r := simplify(combine(rem(p, q, x)));
        p := q;
        q := r;
    end do;

    print(factor(p));
end proc;
```

## ユークリッドの互除法 (ISCZ 行列法)

| 入力引数 | 内容 |
|---|---|
| a | 多項式 A |
| b | 多項式 B |
| n | 精度桁 |

```
isczgeuclid := proc (a, b, n)
    local ia, ib, ipn, dega, degb, che, i, m, kukana, kukanb, sum, sum2, k, c, sho,
            seki, j, degan, amari, sl, sn, st, s, ans, sla, slb, sc, sd, se, eva, pa,
            pb, ku, pu, sk, na, nb, yo, zwc, mey, ss, eyo;

    ia := a;
    ib := b;
    dega := degree(a);
    degb := degree(b);
    sn := 0;
    zwc := 0;
    mey := 0;
    Digits := 2*n;

#区間化
    for i from 0 to dega do
        if coeff(a, x, i) <> 0 then
            if 1 <= abs(evalf(coeff(a, x, i))) then
                m := n+2;
            else
```

```
            m := n;
        end if;
        if type(coeff(a, x, i), fraction) = 'false' then
            m := m+2;
        end if;
        kukana[i, 0] := evalf((round(coeff(a, x, i)*10^n)-0.5)/10^n, m+1);
        kukana[i, 1] := evalf((round(coeff(a, x, i)*10^n)+0.5)/10^n, m+1);
        kukana[i, 2] := sn;
        sl[sn, 0] := coeff(a, x, i);
        sl[sn, 1] := -1;
        sl[sn, 2] := -1;
        sn := sn+1;
    else
        kukana[i, 0] := 0;
        kukana[i, 1] := 0;
        kukana[i, 2] := sn;
        sl[sn, 0] := 0;
        sl[sn, 1] := -1;
        sl[sn, 2] := -1;
        sn := sn+1;
    end if;
end do;

sum := 0;
for k from dega by -1 to 0 do
    if kukana[k, 0]*kukana[k, 1] <> 0 then
        sum := sum+[[kukana[k, 0], kukana[k, 1]], kukana[k, 2]]*x^k;
    end if;
end do;
che := sum;

for i from 0 to degb do
    if coeff(b, x, i) <> 0 then
        if 1 <= abs(evalf(coeff(b, x, i))) then
            m := n+2;
        else
            m := n;
        end if;
        if type(coeff(b, x, i), fraction) = 'false' then
            m := m+2;
        end if;
        kukanb[i, 0] := evalf((round(coeff(b, x, i)*10^n)-0.5)/10^n, m+1);
        kukanb[i, 1] := evalf((round(coeff(b, x, i)*10^n)+0.5)/10^n, m+1);
        kukanb[i, 2] := sn;
        sl[sn, 0] := coeff(b, x, i);
        sl[sn, 1] := -1;
        sl[sn, 2] := -1;
        sn := sn+1;
    else
        kukanb[i, 0] := 0;
        kukanb[i, 1] := 0;
        kukanb[i, 2] := sn;
        sl[sn, 0] := 0;
        sl[sn, 1] := -1;
```

```
            sl[sn, 2] := -1;
            sn := sn+1;
        end if;
    end do;

    sum := 0;
    for k from degb by -1 to 0 do
        if kukanb[k, 0]*kukanb[k, 1] <> 0 then
            sum := sum+[[kukanb[k, 0], kukanb[k, 1]], kukanb[k, 2]]*x^k;
        end if;
    end do;

#ユークリッドの互除法
    while che <> 0 do
        c := -1;
        for i from degb by -1 to 0 do
            if kukanb[i, 0] = 0 then
                c := c+1;
            end if;
        end do;
        if c = degb then
            print(frag0);
            break;
        else
            for i from dega-degb by -1 to 0 do
                sho[i, 0] := min(kukana[degb+i, 0]/kukanb[degb, 0],
                                 kukana[degb+i, 0]/kukanb[degb, 1],
                                 kukana[degb+i, 1]/kukanb[degb, 0],
                                 kukana[degb+i, 1]/kukanb[degb, 1]);
                sho[i, 1] := max(kukana[degb+i, 0]/kukanb[degb, 0],
                                 kukana[degb+i, 0]/kukanb[degb, 1],
                                 kukana[degb+i, 1]/kukanb[degb, 0],
                                 kukana[degb+i, 1]/kukanb[degb, 1]);
                sho[i, 2] := sn;
                sl[sn, 0] := kukana[degb+i, 2];
                sl[sn, 1] := kukanb[degb, 2];
                sl[sn, 2] := "÷";
                sn := sn+1;
                for j from degb by -1 to 0 do
                    seki[j+i, 0] := min(kukanb[j, 0]*sho[i, 0], kukanb[j, 0]*sho[i, 1],
                                        kukanb[j, 1]*sho[i, 0], kukanb[j, 1]*sho[i, 1]);
                    seki[j+i, 1] := max(kukanb[j, 0]*sho[i, 0], kukanb[j, 0]*sho[i, 1],
                                        kukanb[j, 1]*sho[i, 0], kukanb[j, 1]*sho[i, 1]);
                    seki[j+i, 2] := sn;
                    sl[sn, 0] := kukanb[j, 2];
                    sl[sn, 1] := sho[i, 2];
                    sl[sn, 2] := "×";
                    sn := sn+1;
                    kukana[j+i, 0] := kukana[j+i, 0]-seki[j+i, 1];
                    kukana[j+i, 1] := kukana[j+i, 1]-seki[j+i, 0];
                    sl[sn, 0] := kukana[j+i, 2];
                    kukana[j+i, 2] := sn;
                    sl[sn, 1] := seki[j+i, 2];
                    sl[sn, 2] := "−";
```

```
                sn := sn+1;
##ゼロ判定
                if kukana[j+i, 0]*kukana[j+i, 1] <= 0 then
                    zwc := zwc+1;
                    s := kukana[j+i, 2];
                    ss := s;
                    unassign(eva);
                    eva[0, 1] := sl[s, 0];
                    eva[0, 2] := sl[s, 1];
                    eva[0, 3] := sl[s, 2];
                    eva[0, 0] := 1;
                    sc := 1;
                    sd := 0;
                    se := 0;
                    yo := 1;
###行列展開
                    while 0 < sc do
                        eva[sd+1, 0] := 0;
                        while se < eva[sd, 0] do
                            if eva[sd, 3*se+3] <> -1 then
                                pa := eva[sd, 3*se+1];
                                pb := eva[sd, 3*se+2];
                                eva[sd+1, 3*eva[sd+1, 0]+1] := sl[pa, 0];
                                eva[sd+1, 3*eva[sd+1, 0]+2] := sl[pa, 1];
                                eva[sd+1, 3*eva[sd+1, 0]+3] := sl[pa, 2];
                                eva[sd+1, 3*eva[sd+1, 0]+4] := sl[pb, 0];
                                eva[sd+1, 3*eva[sd+1, 0]+5] := sl[pb, 1];
                                eva[sd+1, 3*eva[sd+1, 0]+6] := sl[pb, 2];
                                eva[sd+1, 0] := eva[sd+1, 0]+2;
                                sc := sc+1;
                                if sl[pa, 2] = -1 then
                                    sc := sc-1;
                                end if;
                                if sl[pb, 2] = -1 then
                                    sc := sc-1;
                                end if;
                            end if;
                            se := se+1;
                        end do;
                        sd := sd+1;
                        yo := yo+eva[sd, 0];
                        se := 0;
                    end do;

                    sc := 0;
                    sd := sd-1;
                    eyo := sd+2+3*yo;
                    if mey < eyo then
                        mey := eyo;
                    end if;
###行列計算
                    while 0 <= sd do
                        while se < eva[sd, 0] do
                            if eva[sd, 3*se+3] <> -1 then
```

```
                                    na := eva[sd+1, 6*sc+1];
                                    nb := eva[sd+1, 6*sc+4];
                                    sla := eva[sd, 3*se+1];
                                    sl[sla, 0] := na;
                                    sl[sla, 1] := -1;
                                    sl[sla, 2] := -1;
                                    slb := eva[sd, 3*se+2];
                                    sl[slb, 0] := nb;
                                    sl[slb, 1] := -1;
                                    sl[slb, 2] := -1;
                                    sc := sc+1;
                                    if eva[sd, 3*se+3] = "−" then
                                        eva[sd, 3*se+1] := simplify(combine(na-nb));
                                    end if;
                                    if eva[sd, 3*se+3] = "×" then
                                        eva[sd, 3*se+1] := simplify(combine(na*nb));
                                    end if;
                                    if eva[sd, 3*se+3] = "÷" then
                                        if nb = 0 then
                                            print("Nup", [zwc, sn, mey], n);
                                            isczgeuclid(ia, ib, n+1);
                                            return 0;
                                        end if;
                                        eva[sd, 3*se+1] := simplify(combine(na/nb));
                                    end if;
                                    eva[sd, 3*se+2] := -1;
                                    eva[sd, 3*se+3] := -1;
                                end if;
                                se := se+1;
                            end do;
                            sd := sd-1;
                            sc := 0;
                            se := 0;
                        end do;

                        ans := eva[0, 1];
                        if ans = 0 then
                            kukana[j+i, 0] := 0;
                            kukana[j+i, 1] := 0;
                            sl[ss, 0] := 0;
                            sl[ss, 1] := -1;
                            sl[ss, 2] := -1
                        else
                            print("Nup", [zwc, sn, mey], n);
                            isczgeuclid(ia, ib, n+1);
                            return 0;
                        end if;
                    end if;
                end do;
            end do;

#ユークリッド入れ替え
            degan := dega;
            for i from dega by -1 to 0 do
```

```
            if kukana[i, 0] = 0 then
                degan := degan-1;
            else
                break;
            end if;
        end do;
        if 0 <= degan then
            for i from 0 to degan do
                amari[i, 0] := kukana[i, 0];
                amari[i, 1] := kukana[i, 1];
                amari[i, 2] := kukana[i, 2];
            end do;
            for i from 0 to degb do
                kukana[i, 0] := kukanb[i, 0];
                kukana[i, 1] := kukanb[i, 1];
                kukana[i, 2] := kukanb[i, 2];
            end do;
            dega := degb;
            for i from 0 to dega do
                if i <= degan then
                    kukanb[i, 0] := amari[i, 0];
                    kukanb[i, 1] := amari[i, 1];
                    kukanb[i, 2] := amari[i, 2];
                else
                    kukanb[i, 0] := 0;
                    kukanb[i, 1] := 0;
                end if;
            end do;
            degb := degan;
        end if;
        sum := 0;
        for k from dega by -1 to 0 do
            if kukana[k, 0]*kukana[k, 1] <> 0 then
                sum := sum+[[kukana[k, 0], kukana[k, 1]], kukana[k, 2]]*x^k;
            end if;
        end do;
        che := sum;
    end if;
end do;

#最終出力前シンボル評価
    sum := 0;
    sum2 := 0;
    for k from degb by -1 to 0 do
        if kukanb[k, 0]*kukanb[k, 1] <> 0 then
            sum := sum+[[kukanb[k, 0], kukanb[k, 1]], kukanb[k, 2]]*x^k;
        end if;
        s := kukanb[k, 2];
        ss := s;
        if sl[s, 2] = -1 then
            ans := sl[s, 0];
        else
            unassign(eva);
            eva[0, 1] := sl[s, 0];
```

```
eva[0, 2] := sl[s, 1];
eva[0, 3] := sl[s, 2];
eva[0, 0] := 1;
sc := 1;
sd := 0;
se := 0;
yo := 1;
```

##行列展開

```
while 0 < sc do
    eva[sd+1, 0] := 0;
    while se < eva[sd, 0] do
        if eva[sd, 3*se+3] <> -1 then
            pa := eva[sd, 3*se+1];
            pb := eva[sd, 3*se+2];
            eva[sd+1, 3*eva[sd+1, 0]+1] := sl[pa, 0];
            eva[sd+1, 3*eva[sd+1, 0]+2] := sl[pa, 1];
            eva[sd+1, 3*eva[sd+1, 0]+3] := sl[pa, 2];
            eva[sd+1, 3*eva[sd+1, 0]+4] := sl[pb, 0];
            eva[sd+1, 3*eva[sd+1, 0]+5] := sl[pb, 1];
            eva[sd+1, 3*eva[sd+1, 0]+6] := sl[pb, 2];
            eva[sd+1, 0] := eva[sd+1, 0]+2;
            sc := sc+1;
            if sl[pa, 2] = -1 then
                sc := sc-1;
            end if;
            if sl[pb, 2] = -1 then
                sc := sc-1;
            end if;
        end if;
        se := se+1;
    end do;
    sd := sd+1;
    yo := yo+eva[sd, 0];
    se := 0;
end do;

sc := 0;
sd := sd-1;
eyo := sd+2+3*yo;
if mey < eyo then
    mey := eyo;
end if;
```

##行列計算

```
while 0 <= sd do
    while se < eva[sd, 0] do
        if eva[sd, 3*se+3] <> -1 then
            na := eva[sd+1, 6*sc+1];
            nb := eva[sd+1, 6*sc+4];
            sla := eva[sd, 3*se+1];
            sl[sla, 0] := na;
            sl[sla, 1] := -1;
            sl[sla, 2] := -1;
            slb := eva[sd, 3*se+2];
            sl[slb, 0] := nb;
```

```
                        sl[slb, 1] := -1;
                        sl[slb, 2] := -1;
                        sc := sc+1;
                        if eva[sd, 3*se+3] = "－" then
                            eva[sd, 3*se+1] := simplify(combine(na-nb));
                        end if;
                        if eva[sd, 3*se+3] = "×" then
                            eva[sd, 3*se+1] := simplify(combine(na*nb));
                        end if;
                        if eva[sd, 3*se+3] = "÷" then
                            if nb = 0 then
                                print("Nup", [zwc, sn, mey], n);
                                isczgeuclid(ia, ib, n+1);
                                return 0;
                            end if;
                            eva[sd, 3*se+1] := simplify(combine(na/nb));
                        end if;
                        eva[sd, 3*se+2] := -1;
                        eva[sd, 3*se+3] := -1;
                    end if;
                    se := se+1;
                end do;
                sd := sd-1;
                sc := 0;
                se := 0;
            end do;

            ans := eva[0, 1];
        end if;
        sum2 := sum2+ans*x^k;
    end do;

#最終出力
    print("correct", [zwc, sn, mey], n);
    print("答え");
    print(sum);
    print(sum2);
end proc;
```

## ユークリッドの互除法 (ISCZ 式列法)

| 入力引数 | 内容 |
|---|---|
| a | 多項式 A |
| b | 多項式 B |
| n | 精度桁 |

```
isczseuclid := proc (a, b, n)
    local ia, ib, ipn, dega, degb, che, i, m, kukana, kukanb, sum, sum2, k, c, sho,
            seki, j, degan, amari, sl, sn, st, s, ans, sla, slb, sc, sd, se, eva, pa,
            pb, ku, pu, sk, na, nb, yo, en, el, bn, e, ex, an, ey, ss, zwc, mey, mel, es;
```

```
    ia := a;
    ib := b;
    dega := degree(a);
    degb := degree(b);
    sn := 0;
    zwc := 0;
    mey := 0;
    Digits := 2*n;

#区間化
    for i from 0 to dega do
        if coeff(a, x, i) <> 0 then
            if 1 <= abs(evalf(coeff(a, x, i))) then
                m := n+2;
            else
                m := n;
            end if;
            if type(coeff(a, x, i), fraction) = 'false' then
                m := m+2;
            end if;
            kukana[i, 0] := evalf((round(coeff(a, x, i)*10^n)-.5)/10^n, m+1);
            kukana[i, 1] := evalf((round(coeff(a, x, i)*10^n)+.5)/10^n, m+1);
            kukana[i, 2] := sn;
            sl[sn, 0] := coeff(a, x, i);
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        else
            kukana[i, 0] := 0;
            kukana[i, 1] := 0;
            kukana[i, 2] := sn;
            sl[sn, 0] := 0;
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        end if;
    end do;

    sum := 0;
    for k from dega by -1 to 0 do
        if kukana[k, 0]*kukana[k, 1] <> 0 then
            sum := sum+[[kukana[k, 0], kukana[k, 1]], kukana[k, 2]]*x^k;
        end if;
    end do;
    che := sum;

    for i from 0 to degb do
        if coeff(b, x, i) <> 0 then
            if 1 <= abs(evalf(coeff(b, x, i))) then
                m := n+2;
            else
                m := n;
            end if;
            if type(coeff(b, x, i), fraction) = 'false' then
```

```
            m := m+2;
        end if;
        kukanb[i, 0] := evalf((round(coeff(b, x, i)*10^n)-.5)/10^n, m+1);
        kukanb[i, 1] := evalf((round(coeff(b, x, i)*10^n)+.5)/10^n, m+1);
        kukanb[i, 2] := sn;
        sl[sn, 0] := coeff(b, x, i);
        sl[sn, 1] := -1;
        sl[sn, 2] := -1;
        sn := sn+1;
    else
        kukanb[i, 0] := 0;
        kukanb[i, 1] := 0;
        kukanb[i, 2] := sn;
        sl[sn, 0] := 0;
        sl[sn, 1] := -1;
        sl[sn, 2] := -1;
        sn := sn+1;
    end if;
end do;

sum := 0;
for k from degb by -1 to 0 do
    if kukanb[k, 0]*kukanb[k, 1] <> 0 then
        sum := sum+[[kukanb[k, 0], kukanb[k, 1]], kukanb[k, 2]]*x^k;
    end if;
end do;

#ユークリッドの互除法
    while che <> 0 do
        c := -1;
        for i from degb by -1 to 0 do
            if kukanb[i, 0] = 0 then
                c := c+1;
            end if;
        end do;
        if c = degb then
            print(frag0);
            break;
        else
            for i from dega-degb by -1 to 0 do
                sho[i, 0] := min(kukana[degb+i, 0]/kukanb[degb, 0],
                                 kukana[degb+i, 0]/kukanb[degb, 1],
                                 kukana[degb+i, 1]/kukanb[degb, 0],
                                 kukana[degb+i, 1]/kukanb[degb, 1]);
                sho[i, 1] := max(kukana[degb+i, 0]/kukanb[degb, 0],
                                 kukana[degb+i, 0]/kukanb[degb, 1],
                                 kukana[degb+i, 1]/kukanb[degb, 0],
                                 kukana[degb+i, 1]/kukanb[degb, 1]);
                sho[i, 2] := sn;
                sl[sn, 0] := kukana[degb+i, 2];
                sl[sn, 1] := kukanb[degb, 2];
                sl[sn, 2] := "÷";
                sn := sn+1;
                for j from degb by -1 to 0 do
```

```
                    seki[j+i, 0] := min(kukanb[j, 0]*sho[i, 0], kukanb[j, 0]*sho[i, 1],
                                        kukanb[j, 1]*sho[i, 0], kukanb[j, 1]*sho[i, 1]);
                    seki[j+i, 1] := max(kukanb[j, 0]*sho[i, 0], kukanb[j, 0]*sho[i, 1],
                                        kukanb[j, 1]*sho[i, 0], kukanb[j, 1]*sho[i, 1]);
                    seki[j+i, 2] := sn;
                    sl[sn, 0] := kukanb[j, 2];
                    sl[sn, 1] := sho[i, 2];
                    sl[sn, 2] := "×";
                    sn := sn+1;
                    kukana[j+i, 0] := kukana[j+i, 0]-seki[j+i, 1];
                    kukana[j+i, 1] := kukana[j+i, 1]-seki[j+i, 0];
                    sl[sn, 0] := kukana[j+i, 2];
                    kukana[j+i, 2] := sn;
                    sl[sn, 1] := seki[j+i, 2];
                    sl[sn, 2] := "−";
                    sn := sn+1;
```

##ゼロ判定

```
                    if kukana[j+i, 0]*kukana[j+i, 1] <= 0 then
                        zwc := zwc+1;
                        s := kukana[j+i, 2];
                        ss := s;
                        unassign(eva);
                        eva[0] := sl[s, 2];
                        eva[1] := s;
                        eva[2] := "s";
                        eva[3] := sl[s, 0];
                        eva[4] := "s";
                        eva[5] := sl[s, 1];
                        en := 2;
                        el := 6;
                        mel := 6;
```

###式列展開・計算

```
                        while 0 < en do
                            bn := 0;
                            while bn < el do
                                if eva[bn] = "s" then
                                    if sl[eva[bn+1], 2] = -1 then
                                        eva[bn] := "r";
                                        eva[bn+1] := sl[eva[bn+1], 0];
                                        en := en-1;
                                        if eva[bn-2] = "+"
                                          or eva[bn-2] = "−"
                                          or eva[bn-2] = "×"
                                          or eva[bn-2] = "÷" then
                                            bn := bn-3;
                                        else
                                            bn := bn-5;
                                        end if;
                                        if bn < 0 then
                                            bn := -1;
                                        end if;
                                    else
                                        s := eva[bn+1];
                                        for e from el-1 by -1 to bn+2 do
```

```
                eva[e+4] := eva[e];
            end do;
            eva[bn] := sl[s, 2];
            eva[bn+1] := s;
            eva[bn+2] := "s";
            eva[bn+3] := sl[s, 0];
            eva[bn+4] := "s";
            eva[bn+5] := sl[s, 1];
            en := en+1;
            el := el+4;
            bn := bn-1;
            if mel < el then
                mel := el;
            end if;
        end if;
    else
        if eva[bn+2] = "r" and eva[bn+4] = "r" then
            if eva[bn] = "+" then
                an := simplify(combine(eva[bn+3]+eva[bn+5]));
            end if;
            if eva[bn] = "−" then
                an := simplify(combine(eva[bn+3]-eva[bn+5]));
            end if;
            if eva[bn] = "×" then
                an := simplify(combine(eva[bn+3]*eva[bn+5]));
            end if;
            if eva[bn] = "÷" then
                if eva[bn+5] = 0 then
                    print("Nup", [zwc, sn, mey], n);
                    isczseuclid(ia, ib, n+1);
                    return 0;
                end if;
                an := simplify(combine(eva[bn+3]/eva[bn+5]));
            end if;
            es := eva[bn+1];
            if es <> 0 then
                sl[es, 0] := an;
                sl[es, 1] := -1;
                sl[es, 2] := -1;
                eva[bn] := "r";
                eva[bn+1] := an;
            end if;
            for ey from bn+2 to el-5 do
                eva[ey] := eva[ey+4];
            end do;
            el := el-4;
            if eva[bn-2] = "+"
              or eva[bn-2] = "−"
              or eva[bn-2] = "×"
              or eva[bn-2] = "÷" then
                bn := bn-3;
            else
                bn := bn-5;
            end if;
```

```
                                    if bn < 0 then
                                        bn := -1;
                                    end if;
                                end if;
                            end if;
                            bn := bn+1;
                        end do;
                    end do;

                    ans := eva[1];
                    if mey < mel then
                        mey := mel;
                    end if;
                    if ans = 0 then
                        kukana[j+i, 0] := 0;
                        kukana[j+i, 1] := 0;
                        sl[ss, 0] := 0;
                        sl[ss, 1] := -1;
                        sl[ss, 2] := -1;
                    else
                        print("Nup", [zwc, sn, mey], n);
                        isczseuclid(ia, ib, n+1);
                        return 0;
                    end if;
                end if;
            end do;
        end do;

#ユークリッド入れ替え
        degan := dega;
        for i from dega by -1 to 0 do
            if kukana[i, 0] = 0 then
                degan := degan-1;
            else
                break;
            end if;
        end do;
        if 0 <= degan then
            for i from 0 to degan do
                amari[i, 0] := kukana[i, 0];
                amari[i, 1] := kukana[i, 1];
                amari[i, 2] := kukana[i, 2];
            end do;
            for i from 0 to degb do
                kukana[i, 0] := kukanb[i, 0];
                kukana[i, 1] := kukanb[i, 1];
                kukana[i, 2] := kukanb[i, 2];
            end do;
            dega := degb;
            for i from 0 to dega do
                if i <= degan then
                    kukanb[i, 0] := amari[i, 0];
                    kukanb[i, 1] := amari[i, 1];
                    kukanb[i, 2] := amari[i, 2];
```

```
                    else
                        kukanb[i, 0] := 0;
                        kukanb[i, 1] := 0;
                    end if;
                end do;
                degb := degan;
            end if;
            sum := 0;
            for k from dega by -1 to 0 do
                if kukana[k, 0]*kukana[k, 1] <> 0 then
                    sum := sum+[[kukana[k, 0], kukana[k, 1]], kukana[k, 2]]*x^k;
                end if;
            end do;
            che := sum;
        end if;
    end do;

#最終出力前シンボル評価
    sum := 0;
    sum2 := 0;
    for k from degb by -1 to 0 do
        if kukanb[k, 0]*kukanb[k, 1] <> 0 then
            sum := sum+[[kukanb[k, 0], kukanb[k, 1]], kukanb[k, 2]]*x^k;
        end if;
        s := kukanb[k, 2];
        if sl[s, 2] = -1 then
            ans := sl[s, 0];
        else
            ss := s;
            unassign(eva);
            eva[0] := sl[s, 2];
            eva[1] := s;
            eva[2] := "s";
            eva[3] := sl[s, 0];
            eva[4] := "s";
            eva[5] := sl[s, 1];
            en := 2;
            el := 6;
            mel := 6;
##式列展開・計算
            while 0 < en do
                bn := 0;
                while bn < el do
                    if eva[bn] = "s" then
                        if sl[eva[bn+1], 2] = -1 then
                            eva[bn] := "r";
                            eva[bn+1] := sl[eva[bn+1], 0];
                            en := en-1;
                            if eva[bn-2] = "+" or eva[bn-2] = "－"
                              or eva[bn-2] = "×" or eva[bn-2] = "÷" then
                                bn := bn-3;
                            else
                                bn := bn-5;
                            end if;
```

```
            if bn < 0 then
                bn := -1;
            end if
        else
            s := eva[bn+1];
            for e from el-1 by -1 to bn+2 do
                eva[e+4] := eva[e];
            end do;
            eva[bn] := sl[s, 2];
            eva[bn+1] := s;
            eva[bn+2] := "s";
            eva[bn+3] := sl[s, 0];
            eva[bn+4] := "s";
            eva[bn+5] := sl[s, 1];
            en := en+1;
            el := el+4;
            bn := bn-1;
            if mel < el then
                mel := el;
            end if;
        end if;
    else
        if eva[bn+2] = "r" and eva[bn+4] = "r" then
            if eva[bn] = "+" then
                an := simplify(combine(eva[bn+3]+eva[bn+5]));
            end if;
            if eva[bn] = "−" then
                an := simplify(combine(eva[bn+3]-eva[bn+5]));
            end if;
            if eva[bn] = "×" then
                an := simplify(combine(eva[bn+3]*eva[bn+5]));
            end if;
            if eva[bn] = "÷" then
                if eva[bn+5] = 0 then
                    print("Nup", [zwc, sn, mey], n);
                    isczseuclid(ia, ib, n+1);
                    return 0;
                end if;
                an := simplify(combine(eva[bn+3]/eva[bn+5]));
            end if;
            es := eva[bn+1];
            if es <> 0 then
                sl[es, 0] := an;
                sl[es, 1] := -1;
                sl[es, 2] := -1;
                eva[bn] := "r";
                eva[bn+1] := an;
            end if;
            for ey from bn+2 to el-5 do
                eva[ey] := eva[ey+4];
            end do;
            el := el-4;
            if eva[bn-2] = "+" or eva[bn-2] = "−"
              or eva[bn-2] = "×" or eva[bn-2] = "÷" then
```

```
                                bn := bn-3;
                            else
                                bn := bn-5;
                            end if;
                            if bn < 0 then
                                bn := -1;
                            end if;
                        end if;
                    end if;
                    bn := bn+1;
                end do;
            end do;

            ans := eva[1];
        end if;
        sum2 := sum2+ans*x^k;
    end do;

#最終出力
    print("correct", [zwc, sn, mey], n);
    print("答え");
    print(sum);
    print(sum2);
end proc;
```

## スツルムアルゴリズム用多項式生成プログラム

| 入力引数 | 内容 |
|---|---|
| n | 実解数 |
| op | オプション (0/有理数のみ 1/無理数あり) |

```
setpoly := proc (n, op)
    local dig, bo, shi, poly, fugo, mr, m;

    poly := 1;
    for dig to n do
        bo := floor(mod(rand(), 10)+1);
        shi := floor(mod(rand(), 10)+1);
        if mod(rand(), 2) = 0 then
            fugo := 1;
        else
            fugo := -1;
        end if;
        mr := rand();
        if mod(mr, 20) = 0 then
            m := 3;
        elif mod(mr, 5) = 0 then
            m := 2;
        else
            m := 1;
        end if;
        if op = 1 then
```

```
            if mod(rand(), 6) = 0 then
                fugo := fugo*sqrt(mod(rand(), 20)+1);
            end if;
            if mod(rand(), 10) = 0 then
                fugo := fugo*Pi;
            end if;
            if mod(rand(), 10) = 0 then
                fugo := fugo*exp(1);
            end if;
        end if;
        print(fugo, bo, shi, m);
        poly := sort(simplify(expand(simplify(poly*(x-fugo*shi/bo)^m))));
    end do;
    print(poly);
end proc;
```

## スツルムアルゴリズム (厳密計算)

| 入力引数 | 内容 |
|---|---|
| a | 多項式 A |
| shita | 解が存在するか調べる下限 |
| ue | 解が存在するか調べる上限 |

```
strm := proc (a, shita, ue)
    local ia, p, n, solche, i, strmans, strmbtm, strmtop;

    ia := a;
#上限, 下限で割り切れるか判定
    solche := 1;
    while solche = 1 do
        solche := 0;
        if algsubs(x = shita, ia) = 0 then
            ia := quo(ia, x-shita, x);
            solche := 1;
        end if;
        if algsubs(x = ue, ia) = 0 then
            ia := quo(ia, x-ue, x);
            solche := 1;
        end if;
    end do;

#スツルム列生成
    p[0, 0] := ia;
    p[0, 1] := signum(algsubs(x = shita, p[0, 0]));
    p[0, 2] := signum(algsubs(x = ue, p[0, 0]));
    p[1, 0] := diff(ia, x);
    p[1, 1] := signum(algsubs(x = shita, p[1, 0]));
    p[1, 2] := signum(algsubs(x = ue, p[1, 0]));
    n := 2;
    while p[n-1, 0] <> 0 do
        p[n, 0] := -rem(p[n-2, 0], p[n-1, 0], x);
```

```
        p[n, 1] := signum(algsubs(x = shita, p[n, 0]));
        p[n, 2] := signum(algsubs(x = ue, p[n, 0]));
        print(p[n-2, 0], p[n-1, 0], p[n, 0]);
        n := n+1;
    end do;

#実根数判定
    strmbtm := 0;
    strmtop := 0;
    for i from 0 to n-1 do
        if p[i, 1] = 0 then
            p[i, 1] := p[i-1, 1];
        end if;
        if p[i, 2] = 0 then
            p[i, 2] := p[i-1, 2];
        end if;
        if 1 <= i then
            if p[i-1, 1] <> p[i, 1] then
                strmbtm := strmbtm+1;
            end if;
            if p[i-1, 2] <> p[i, 2] then
                strmtop := strmtop+1;
            end if;
        end if;
    end do;
    strmans := abs(strmbtm-strmtop);

#出力
    print("答え");
    print(strmans);
end proc;
```

## スツルムアルゴリズム (ISCZ 行列法)

| 入力引数 | 内容 |
|---|---|
| a | 多項式 A |
| shita | 解が存在するか調べる下限 |
| ue | 解が存在するか調べる上限 |
| n | 精度桁 |

```
isczgstrm := proc (a, shita, ue, n)
    local ia, b, ib, ipn, dega, degb, remcount, che, i, m, kukan, sum, sum2, sum3, k, c,
            sho, seki, j, degan, amari, sl, sn, st, s, ans, sla, slb, sc, sd, se, eva,
            pa, pb, ku, pu, sk, na, nb, yo, zwc, mey, ss, eyo, fugo, strmbtm, strmtop,
            strmans, stac, solche;

    ia := a;
#上限, 下限で割り切れるか判定
    solche := 1;
    while solche = 1 do
        solche := 0;
```

```
        if algsubs(x = shita, ia) = 0 then
            ia := quo(ia, x-shita, x);
            solche := 1;
        end if;
        if algsubs(x = ue, ia) = 0 then
            ia := quo(ia, x-ue, x);
            solche := 1;
        end if;
    end do;

#スツルム列生成 (1)
    b := diff(ia, x);
    ib := b;
    dega := degree(ia);
    degb := degree(b);
    sn := 0;
    zwc := 0;
    mey := 0;
    fugo[0, 0] := ia;
    fugo[0, 1] := signum(algsubs(x = shita, ia));
    fugo[0, 2] := signum(algsubs(x = ue, ia));
    fugo[1, 0] := b;
    fugo[1, 1] := signum(algsubs(x = shita, b));
    fugo[1, 2] := signum(algsubs(x = ue, b));
    sl[0, 0] := -1;
    sl[0, 1] := -1;
    sl[0, 2] := -1;
    sn := sn+1;

#区間化
    for i from 0 to dega do
        if coeff(ia, x, i) <> 0 then
            if 1 <= abs(evalf(coeff(ia, x, i))) then
                m := n+2;
            else
                m := n;
            end if;
            if type(coeff(ia, x, i), fraction) = 'false' then
                m := m+2
            end if;
            kukan[0, i, 0] := evalf(floor(coeff(ia, x, i)*10^n)/10^n, m);
            kukan[0, i, 1] := evalf(ceil(coeff(ia, x, i)*10^n)/10^n, m);
            kukan[0, i, 2] := sn;
            sl[sn, 0] := coeff(ia, x, i);
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        else
            kukan[0, i, 0] := 0;
            kukan[0, i, 1] := 0;
            kukan[0, i, 2] := sn;
            sl[sn, 0] := 0;
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
```

```
            sn := sn+1;
        end if;
    end do;

    sum := 0;
    for k from dega by -1 to 0 do
        if kukan[0, k, 0]*kukan[0, k, 1] <> 0 then
            sum := sum+[[kukan[0, k, 0], kukan[0, k, 1]], kukan[0, k, 2]]*x^k;
        end if;
    end do;
    che := sum;

    for i from 0 to degb do
        if coeff(b, x, i) <> 0 then
            if 1 <= abs(evalf(coeff(b, x, i))) then
                m := n+2;
            else
                m := n;
            end if;
            if type(coeff(b, x, i), fraction) = 'false' then
                m := m+2;
            end if;
            kukan[1, i, 0] := evalf(floor(coeff(b, x, i)*10^n)/10^n, m);
            kukan[1, i, 1] := evalf(ceil(coeff(b, x, i)*10^n)/10^n, m);
            kukan[1, i, 2] := sn;
            sl[sn, 0] := coeff(b, x, i);
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        else
            kukan[1, i, 0] := 0;
            kukan[1, i, 1] := 0;
            kukan[1, i, 2] := sn;
            sl[sn, 0] := 0;
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        end if;
    end do;

#スツルム列生成 (2)
    remcount := 2;
    while che <> 0 do
        c := -1;
        for i from degb by -1 to 0 do
            if kukan[remcount-1, i, 0] = 0 then
                c := c+1;
            end if;
        end do;
        for i from dega by -1 to 0 do
            kukan[remcount, i, 0] := kukan[remcount-2, i, 0];
            kukan[remcount, i, 1] := kukan[remcount-2, i, 1];
            kukan[remcount, i, 2] := kukan[remcount-2, i, 2];
        end do;
```

```
        if c = degb then
            print(frag0);
            break;
        else
            for i from dega-degb by -1 to 0 do
                sho[i, 0] := min(kukan[remcount, degb+i, 0]/kukan[remcount-1, degb, 0],
                                 kukan[remcount, degb+i, 0]/kukan[remcount-1, degb, 1],
                                 kukan[remcount, degb+i, 1]/kukan[remcount-1, degb, 0],
                                 kukan[remcount, degb+i, 1]/kukan[remcount-1, degb, 1]);
                sho[i, 1] := max(kukan[remcount, degb+i, 0]/kukan[remcount-1, degb, 0],
                                 kukan[remcount, degb+i, 0]/kukan[remcount-1, degb, 1],
                                 kukan[remcount, degb+i, 1]/kukan[remcount-1, degb, 0],
                                 kukan[remcount, degb+i, 1]/kukan[remcount-1, degb, 1]);
                sho[i, 2] := sn;
                sl[sn, 0] := kukan[remcount, degb+i, 2];
                sl[sn, 1] := kukan[remcount-1, degb, 2];
                sl[sn, 2] := "÷";
                sn := sn+1;
                for j from degb by -1 to 0 do
                    seki[j+i, 0] := min(kukan[remcount-1, j, 0]*sho[i, 0],
                                        kukan[remcount-1, j, 0]*sho[i, 1],
                                        kukan[remcount-1, j, 1]*sho[i, 0],
                                        kukan[remcount-1, j, 1]*sho[i, 1]);
                    seki[j+i, 1] := max(kukan[remcount-1, j, 0]*sho[i, 0],
                                        kukan[remcount-1, j, 0]*sho[i, 1],
                                        kukan[remcount-1, j, 1]*sho[i, 0],
                                        kukan[remcount-1, j, 1]*sho[i, 1]);
                    seki[j+i, 2] := sn;
                    sl[sn, 0] := kukan[remcount-1, j, 2];
                    sl[sn, 1] := sho[i, 2];
                    sl[sn, 2] := "×";
                    sn := sn+1;
                    kukan[remcount, j+i, 0] := kukan[remcount, j+i, 0]-seki[j+i, 1];
                    kukan[remcount, j+i, 1] := kukan[remcount, j+i, 1]-seki[j+i, 0];
                    sl[sn, 0] := kukan[remcount, j+i, 2];
                    kukan[remcount, j+i, 2] := sn;
                    sl[sn, 1] := seki[j+i, 2];
                    sl[sn, 2] := "−";
                    sn := sn+1;
##ゼロ判定
                    if kukan[remcount, j+i, 0]*kukan[remcount, j+i, 1] <= 0 then
                        zwc := zwc+1;
                        s := kukan[remcount, j+i, 2];
                        ss := s;
                        unassign(eva);
                        eva[0, 1] := sl[s, 0];
                        eva[0, 2] := sl[s, 1];
                        eva[0, 3] := sl[s, 2];
                        eva[0, 0] := 1;
                        sc := 1;
                        sd := 0;
                        se := 0;
                        yo := 1;
###行列展開
```

```
while 0 < sc do
    eva[sd+1, 0] := 0;
    while se < eva[sd, 0] do
        if eva[sd, 3*se+3] <> -1 then
            pa := eva[sd, 3*se+1];
            pb := eva[sd, 3*se+2];
            eva[sd+1, 3*eva[sd+1, 0]+1] := sl[pa, 0];
            eva[sd+1, 3*eva[sd+1, 0]+2] := sl[pa, 1];
            eva[sd+1, 3*eva[sd+1, 0]+3] := sl[pa, 2];
            eva[sd+1, 3*eva[sd+1, 0]+4] := sl[pb, 0];
            eva[sd+1, 3*eva[sd+1, 0]+5] := sl[pb, 1];
            eva[sd+1, 3*eva[sd+1, 0]+6] := sl[pb, 2];
            eva[sd+1, 0] := eva[sd+1, 0]+2;
            sc := sc+1;
            if sl[pa, 2] = -1 then
                sc := sc-1;
            end if;
            if sl[pb, 2] = -1 then
                sc := sc-1;
            end if;
        end if;
        se := se+1;
    end do;
    sd := sd+1;
    yo := yo+eva[sd, 0];
    se := 0;
end do;
sc := 0;
sd := sd-1;
eyo := sd+2+3*yo;
if mey < eyo then
    mey := eyo;
end if;
```

###行列計算

```
while 0 <= sd do
    while se < eva[sd, 0] do
        if eva[sd, 3*se+3] <> -1 then
            na := eva[sd+1, 6*sc+1];
            nb := eva[sd+1, 6*sc+4];
            sla := eva[sd, 3*se+1];
            sl[sla, 0] := na;
            sl[sla, 1] := -1;
            sl[sla, 2] := -1;
            slb := eva[sd, 3*se+2];
            sl[slb, 0] := nb;
            sl[slb, 1] := -1;
            sl[slb, 2] := -1;
            sc := sc+1;
            if eva[sd, 3*se+3] = "－" then
                eva[sd, 3*se+1] := simplify(combine(na-nb));
            end if;
            if eva[sd, 3*se+3] = "×" then
                eva[sd, 3*se+1] := simplify(combine(na*nb));
            end if;
```

```
                                    if eva[sd, 3*se+3] = "÷" then
                                        if nb = 0 then
                                            print("Nup", [zwc, sn, mey], n);
                                            isczgstrm(ia, shita, ue, n+1);
                                            return 0;
                                        end if;
                                        eva[sd, 3*se+1] := simplify(combine(na/nb));
                                    end if;
                                    eva[sd, 3*se+2] := -1;
                                    eva[sd, 3*se+3] := -1;
                                end if;
                                se := se+1;
                            end do;
                            sd := sd-1;
                            sc := 0;
                            se := 0;
                        end do;

                        ans := eva[0, 1];
                        if ans = 0 then
                            kukan[remcount, j+i, 0] := 0;
                            kukan[remcount, j+i, 1] := 0;
                            sl[ss, 0] := 0;
                            sl[ss, 1] := -1;
                            sl[ss, 2] := -1;
                        else
                            print("Nup", [zwc, sn, mey], n);
                            isczgstrm(ia, shita, ue, n+1);
                            return 0;
                        end if;
                    end if;
                end do;
            end do;
```

##入れ替え

```
            degan := dega;
            for i from dega by -1 to 0 do
                if kukan[remcount, i, 0] = 0 then
                    degan := degan-1;
                else
                    break;
                end if;
            end do;
            for k from 0 to degan do
                stac := -kukan[remcount, k, 0];
                kukan[remcount, k, 0] := -kukan[remcount, k, 1];
                kukan[remcount, k, 1] := stac;
                sl[sn, 0] := kukan[remcount, k, 2];
                kukan[remcount, k, 2] := sn;
                sl[sn, 1] := 0;
                sl[sn, 2] := "×";
                sn := sn+1;
            end do;
            if 0 <= degan then
```

```
            dega := degb;
            degb := 0;
            remcount := remcount+1;
        else
            break;
        end if;

        sum := 0;
        sum2 := 0;
        sum3 := 0;
        for k from dega by -1 to 0 do
###ゼロ判定
            if kukan[remcount-1, k, 0]*kukan[remcount-1, k, 1] <> 0 then
                sum := sum+[[kukan[remcount-1, k, 0], kukan[remcount-1, k, 1]],
                                                        kukan[remcount-1, k, 2]]*x^k;
                sum2 := sum2+kukan[remcount-1, k, 0]*x^k;
                if sl[kukan[remcount-1, k, 2], 2] <> -1 then
                    zwc := zwc+1;
                    s := kukan[remcount-1, k, 2];
                    ss := s;
                    unassign(eva);
                    eva[0, 1] := sl[s, 0];
                    eva[0, 2] := sl[s, 1];
                    eva[0, 3] := sl[s, 2];
                    eva[0, 0] := 1;
                    sc := 1;
                    sd := 0;
                    se := 0;
                    yo := 1;
####行列展開
                    while 0 < sc do
                        eva[sd+1, 0] := 0;
                        while se < eva[sd, 0] do
                            if eva[sd, 3*se+3] <> -1 then
                                pa := eva[sd, 3*se+1];
                                pb := eva[sd, 3*se+2];
                                eva[sd+1, 3*eva[sd+1, 0]+1] := sl[pa, 0];
                                eva[sd+1, 3*eva[sd+1, 0]+2] := sl[pa, 1];
                                eva[sd+1, 3*eva[sd+1, 0]+3] := sl[pa, 2];
                                eva[sd+1, 3*eva[sd+1, 0]+4] := sl[pb, 0];
                                eva[sd+1, 3*eva[sd+1, 0]+5] := sl[pb, 1];
                                eva[sd+1, 3*eva[sd+1, 0]+6] := sl[pb, 2];
                                eva[sd+1, 0] := eva[sd+1, 0]+2;
                                sc := sc+1;
                                if sl[pa, 2] = -1 then
                                    sc := sc-1;
                                end if;
                                if sl[pb, 2] = -1 then
                                    sc := sc-1;
                                end if;
                            end if;
                            se := se+1;
                        end do;
                        sd := sd+1;
```

```
            yo := yo+eva[sd, 0];
            se := 0;
        end do;
        sc := 0;
        sd := sd-1;
        eyo := sd+2+3*yo;
        if mey < eyo then
            mey := eyo;
        end if;
####行列計算
        while 0 <= sd do
            while se < eva[sd, 0] do
                if eva[sd, 3*se+3] <> -1 then
                    na := eva[sd+1, 6*sc+1];
                    nb := eva[sd+1, 6*sc+4];
                    sla := eva[sd, 3*se+1];
                    sl[sla, 0] := na;
                    sl[sla, 1] := -1;
                    sl[sla, 2] := -1;
                    slb := eva[sd, 3*se+2];
                    sl[slb, 0] := nb;
                    sl[slb, 1] := -1;
                    sl[slb, 2] := -1;
                    sc := sc+1;
                    if eva[sd, 3*se+3] = "－" then
                        eva[sd, 3*se+1] := simplify(combine(na-nb));
                    end if;
                    if eva[sd, 3*se+3] = "×" then
                        eva[sd, 3*se+1] := simplify(combine(na*nb));
                    end if;
                    if eva[sd, 3*se+3] = "÷" then
                        if nb = 0 then
                            print("Nup", [zwc, sn, mey], n);
                            isczgstrm(ia, shita, ue, n+1);
                            return 0;
                        end if;
                        eva[sd, 3*se+1] := simplify(combine(na/nb));
                    end if;
                    eva[sd, 3*se+2] := -1;
                    eva[sd, 3*se+3] := -1;
                end if;
                se := se+1;
            end do;
            sd := sd-1;
            sc := 0;
            se := 0;
        end do;

        ans := eva[0, 1];
    else
        ans := sl[kukan[remcount-1, k, 2], 0];
    end if;
    sum3 := sum3+ans*x^k;
    if degb < k then
```

```
                    degb := k;
                end if;
            end if;
        end do;
        fugo[remcount-1, 0] := sum3;
        fugo[remcount-1, 1] := signum(algsubs(x = shita, sum3));
        fugo[remcount-1, 2] := signum(algsubs(x = ue, sum3));
        che := degb;
    end if;
end do;

#実解数計算
    strmbtm := 0;
    strmtop := 0;
    for i from 0 to remcount-1 do
        if fugo[i, 1] = 0 then
            fugo[i, 1] := fugo[i-1, 1];
        end if;
        if fugo[i, 2] = 0 then
            fugo[i, 2] := fugo[i-1, 2];
        end if;
        if 1 <= i then
            if fugo[i-1, 1] <> fugo[i, 1] then
                strmbtm := strmbtm+1;
            end if;
            if fugo[i-1, 2] <> fugo[i, 2] then
                strmtop := strmtop+1;
            end if;
        end if;
    end do;

#最終出力
    strmans := abs(strmbtm-strmtop);
    print("correct", [zwc, sn, mey], n);
    print("答え");
    print(strmans);
end proc;
```

## スツルムアルゴリズム (ISCZ 式列法)

| 入力引数 | 内容 |
|---|---|
| a | 多項式 A |
| shita | 解が存在するか調べる下限 |
| ue | 解が存在するか調べる上限 |
| n | 精度桁 |

```
isczsstrm := proc (a, shita, ue, n)
    local ia, b, ib, ipn, dega, degb, remcount, che, i, m, kukan, sum, sum2, sum3, k, c,
            sho, seki, j, degan, amari, sl, sn, st, s, ans, sla, slb, sc, sd, se, eva,
            pa, pb, ku, pu, sk, na, nb, yo, zwc, mey, ss, eyo, fugo, strmbtm, strmtop,
            strmans, stac, solche, en, el, mel, e, an, bn, es, ey;
```

```
    ia := a;
#上限, 下限で割り切れるか判定
    solche := 1;
    while solche = 1 do
        solche := 0;
        if algsubs(x = shita, ia) = 0 then
            ia := quo(ia, x-shita, x);
            solche := 1;
        end if;
        if algsubs(x = ue, ia) = 0 then
            ia := quo(ia, x-ue, x);
            solche := 1;
        end if;
    end do;

#スツルム列生成 (1)
    b := diff(ia, x);
    ib := b;
    dega := degree(ia);
    degb := degree(b);
    sn := 0;
    zwc := 0;
    mey := 0;
    fugo[0, 0] := ia;
    fugo[0, 1] := signum(algsubs(x = shita, ia));
    fugo[0, 2] := signum(algsubs(x = ue, ia));
    fugo[1, 0] := b;
    fugo[1, 1] := signum(algsubs(x = shita, b));
    fugo[1, 2] := signum(algsubs(x = ue, b));
    sl[0, 0] := -1;
    sl[0, 1] := -1;
    sl[0, 2] := -1;
    sn := sn+1;

#区間化
    for i from 0 to dega do
        if coeff(ia, x, i) <> 0 then
            if 1 <= abs(evalf(coeff(ia, x, i))) then
                m := n+2;
            else
                m := n;
            end if;
            if type(coeff(ia, x, i), fraction) = 'false' then
                m := m+2
            end if;
            kukan[0, i, 0] := evalf(floor(coeff(ia, x, i)*10^n)/10^n, m);
            kukan[0, i, 1] := evalf(ceil(coeff(ia, x, i)*10^n)/10^n, m);
            kukan[0, i, 2] := sn;
            sl[sn, 0] := coeff(ia, x, i);
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        else
```

```
            kukan[0, i, 0] := 0;
            kukan[0, i, 1] := 0;
            kukan[0, i, 2] := sn;
            sl[sn, 0] := 0;
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        end if;
    end do;

    sum := 0;
    for k from dega by -1 to 0 do
        if kukan[0, k, 0]*kukan[0, k, 1] <> 0 then
            sum := sum+[[kukan[0, k, 0], kukan[0, k, 1]], kukan[0, k, 2]]*x^k;
        end if;
    end do;
    che := sum;

    for i from 0 to degb do
        if coeff(b, x, i) <> 0 then
            if 1 <= abs(evalf(coeff(b, x, i))) then
                m := n+2;
            else
                m := n;
            end if;
            if type(coeff(b, x, i), fraction) = 'false' then
                m := m+2;
            end if;
            kukan[1, i, 0] := evalf(floor(coeff(b, x, i)*10^n)/10^n, m);
            kukan[1, i, 1] := evalf(ceil(coeff(b, x, i)*10^n)/10^n, m);
            kukan[1, i, 2] := sn;
            sl[sn, 0] := coeff(b, x, i);
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        else
            kukan[1, i, 0] := 0;
            kukan[1, i, 1] := 0;
            kukan[1, i, 2] := sn;
            sl[sn, 0] := 0;
            sl[sn, 1] := -1;
            sl[sn, 2] := -1;
            sn := sn+1;
        end if;
    end do;

#スツルム列生成 (2)
    remcount := 2;
    while che <> 0 do
        c := -1;
        for i from degb by -1 to 0 do
            if kukan[remcount-1, i, 0] = 0 then
                c := c+1;
            end if;
```

```
end do;
for i from dega by -1 to 0 do
    kukan[remcount, i, 0] := kukan[remcount-2, i, 0];
    kukan[remcount, i, 1] := kukan[remcount-2, i, 1];
    kukan[remcount, i, 2] := kukan[remcount-2, i, 2];
end do;
if c = degb then
    print(frag0);
    break;
else
    for i from dega-degb by -1 to 0 do
        sho[i, 0] := min(kukan[remcount, degb+i, 0]/kukan[remcount-1, degb, 0],
                         kukan[remcount, degb+i, 0]/kukan[remcount-1, degb, 1],
                         kukan[remcount, degb+i, 1]/kukan[remcount-1, degb, 0],
                         kukan[remcount, degb+i, 1]/kukan[remcount-1, degb, 1]);
        sho[i, 1] := max(kukan[remcount, degb+i, 0]/kukan[remcount-1, degb, 0],
                         kukan[remcount, degb+i, 0]/kukan[remcount-1, degb, 1],
                         kukan[remcount, degb+i, 1]/kukan[remcount-1, degb, 0],
                         kukan[remcount, degb+i, 1]/kukan[remcount-1, degb, 1]);
        sho[i, 2] := sn;
        sl[sn, 0] := kukan[remcount, degb+i, 2];
        sl[sn, 1] := kukan[remcount-1, degb, 2];
        sl[sn, 2] := "÷";
        sn := sn+1;
        for j from degb by -1 to 0 do
            seki[j+i, 0] := min(kukan[remcount-1, j, 0]*sho[i, 0],
                                kukan[remcount-1, j, 0]*sho[i, 1],
                                kukan[remcount-1, j, 1]*sho[i, 0],
                                kukan[remcount-1, j, 1]*sho[i, 1]);
            seki[j+i, 1] := max(kukan[remcount-1, j, 0]*sho[i, 0],
                                kukan[remcount-1, j, 0]*sho[i, 1],
                                kukan[remcount-1, j, 1]*sho[i, 0],
                                kukan[remcount-1, j, 1]*sho[i, 1]);
            seki[j+i, 2] := sn;
            sl[sn, 0] := kukan[remcount-1, j, 2];
            sl[sn, 1] := sho[i, 2];
            sl[sn, 2] := "×";
            sn := sn+1;
            kukan[remcount, j+i, 0] := kukan[remcount, j+i, 0]-seki[j+i, 1];
            kukan[remcount, j+i, 1] := kukan[remcount, j+i, 1]-seki[j+i, 0];
            sl[sn, 0] := kukan[remcount, j+i, 2];
            kukan[remcount, j+i, 2] := sn;
            sl[sn, 1] := seki[j+i, 2];
            sl[sn, 2] := "−";
            sn := sn+1;
##ゼロ判定
            if kukan[remcount, j+i, 0]*kukan[remcount, j+i, 1] <= 0 then
                zwc := zwc+1;
                s := kukan[remcount, j+i, 2];
                ss := s;
                unassign(eva);
                eva[0] := sl[s, 2];
                eva[1] := s;
                eva[2] := "s";
```

```
eva[3] := sl[s, 0];
eva[4] := "s";
eva[5] := sl[s, 1];
en := 2;
el := 6;
mel := 6;
```

###式列展開・計算

```
while 0 < en do
    bn := 0;
    while bn < el do
        if eva[bn] = "s" then
            if sl[eva[bn+1], 2] = -1 then
                eva[bn] := "r";
                eva[bn+1] := sl[eva[bn+1], 0];
                en := en-1;
                if eva[bn-2] = "+"
                  or eva[bn-2] = "−"
                  or eva[bn-2] = "×"
                  or eva[bn-2] = "÷" then
                    bn := bn-3;
                else
                    bn := bn-5;
                end if;
                if bn < 0 then
                    bn := -1;
                end if;
            else
                s := eva[bn+1];
                for e from el-1 by -1 to bn+2 do
                    eva[e+4] := eva[e];
                end do;
                eva[bn] := sl[s, 2];
                eva[bn+1] := s;
                eva[bn+2] := "s";
                eva[bn+3] := sl[s, 0];
                eva[bn+4] := "s";
                eva[bn+5] := sl[s, 1];
                en := en+1;
                el := el+4;
                bn := bn-1;
                if mel < el then
                    mel := el;
                end if;
            end if;
        else
            if eva[bn+2] = "r" and eva[bn+4] = "r" then
                if eva[bn] = "−" then
                    an := simplify(combine(eva[bn+3]-eva[bn+5]));
                end if;
                if eva[bn] = "×" then
                    an := simplify(combine(eva[bn+3]*eva[bn+5]));
                end if;
                if eva[bn] = "÷" then
                    if eva[bn+5] = 0 then
```

```
                                        print("Nup", [zwc, sn, mey], n);
                                        kukaneuclid(ia, ib, n+1);
                                        return 0;
                                    end if;
                                    an := simplify(combine(eva[bn+3]/eva[bn+5]));
                                end if;
                                es := eva[bn+1];
                                if es <> 0 then
                                    sl[es, 0] := an;
                                    sl[es, 1] := -1;
                                    sl[es, 2] := -1;
                                    eva[bn] := "r";
                                    eva[bn+1] := an;
                                end if;
                                for ey from bn+2 to el-5 do
                                    eva[ey] := eva[ey+4];
                                end do;
                                el := el-4;
                                if eva[bn-2] = "+"
                                  or eva[bn-2] = "−"
                                  or eva[bn-2] = "×"
                                  or eva[bn-2] = "÷" then
                                    bn := bn-3;
                                else
                                    bn := bn-5;
                                end if;
                                if bn < 0 then
                                    bn := -1;
                                end if;
                            end if;
                        end if;
                        bn := bn+1;
                    end do;
                end do;

                ans := eva[1];
                if mey < mel then
                    mey := mel;
                end if;
                if ans = 0 then
                    kukan[remcount, j+i, 0] := 0;
                    kukan[remcount, j+i, 1] := 0;
                    sl[ss, 0] := 0;
                    sl[ss, 1] := -1;
                    sl[ss, 2] := -1;
                else
                    print("Nup", [zwc, sn, mey], n);
                    isczsstrm(ia, shita, ue, n+1);
                    return 0;
                end if;
            end if;
        end do;
    end do;
```

##入れ替え

```
            degan := dega;
            for i from dega by -1 to 0 do
                if kukan[remcount, i, 0] = 0 then
                    degan := degan-1;
                else
                    break;
                end if;
            end do;
            for k from 0 to degan do
                stac := -kukan[remcount, k, 0];
                kukan[remcount, k, 0] := -kukan[remcount, k, 1];
                kukan[remcount, k, 1] := stac;
                sl[sn, 0] := kukan[remcount, k, 2];
                kukan[remcount, k, 2] := sn;
                sl[sn, 1] := 0;
                sl[sn, 2] := "×";
                sn := sn+1;
            end do;
            if 0 <= degan then
                dega := degb;
                degb := 0;
                remcount := remcount+1;
            else
                break;
            end if;

            sum := 0;
            sum2 := 0;
            sum3 := 0;
            for k from dega by -1 to 0 do
```

###ゼロ判定

```
                if kukan[remcount-1, k, 0]*kukan[remcount-1, k, 1] <> 0 then
                    sum := sum+[[kukan[remcount-1, k, 0], kukan[remcount-1, k, 1]],
                                                          kukan[remcount-1, k, 2]]*x^k;
                    sum2 := sum2+kukan[remcount-1, k, 0]*x^k;
                    if sl[kukan[remcount-1, k, 2], 2] <> -1 then
                        zwc := zwc+1;
                        s := kukan[remcount-1, k, 2];
                        ss := s;
                        unassign(eva);
                        eva[0] := sl[s, 2];
                        eva[1] := s;
                        eva[2] := "s";
                        eva[3] := sl[s, 0];
                        eva[4] := "s";
                        eva[5] := sl[s, 1];
                        en := 2;
                        el := 6;
                        mel := 6;
```

####式列展開・計算

```
                        while 0 < en do
                            bn := 0;
                            while bn < el do
```

```
if eva[bn] = "s" then
    if sl[eva[bn+1], 2] = -1 then
        eva[bn] := "r";
        eva[bn+1] := sl[eva[bn+1], 0];
        en := en-1;
        if eva[bn-2] = "+"
          or eva[bn-2] = "−"
          or eva[bn-2] = "×"
          or eva[bn-2] = "÷" then
            bn := bn-3;
        else
            bn := bn-5;
        end if;
        if bn < 0 then
            bn := -1;
        end if;
    else
        s := eva[bn+1];
        for e from el-1 by -1 to bn+2 do
            eva[e+4] := eva[e];
        end do;
        eva[bn] := sl[s, 2];
        eva[bn+1] := s;
        eva[bn+2] := "s";
        eva[bn+3] := sl[s, 0];
        eva[bn+4] := "s";
        eva[bn+5] := sl[s, 1];
        en := en+1;
        el := el+4;
        bn := bn-1;
        if mel < el then
            mel := el;
        end if;
    end if;
else
    if eva[bn+2] = "r" and eva[bn+4] = "r" then
        if eva[bn] = "−" then
            an := simplify(combine(eva[bn+3]-eva[bn+5]));
        end if;
        if eva[bn] = "×" then
            an := simplify(combine(eva[bn+3]*eva[bn+5]));
        end if;
        if eva[bn] = "÷" then
            if eva[bn+5] = 0 then
                print("Nup", [zwc, sn, mey], n);
                isczsstrm(ia, ib, n+1);
                return 0;
            end if;
            an := simplify(combine(eva[bn+3]/eva[bn+5]));
        end if;
        es := eva[bn+1];
        if es <> 0 then
            sl[es, 0] := an;
            sl[es, 1] := -1;
```

```
                                        sl[es, 2] := -1;
                                        eva[bn] := "r";
                                        eva[bn+1] := an;
                                    end if;
                                    for ey from bn+2 to el-5 do
                                        eva[ey] := eva[ey+4];
                                    end do;
                                    el := el-4;
                                    if eva[bn-2] = "+"
                                      or eva[bn-2] = "−"
                                      or eva[bn-2] = "×"
                                      or eva[bn-2] = "÷" then
                                        bn := bn-3;
                                    else
                                        bn := bn-5;
                                    end if;
                                    if bn < 0 then
                                        bn := -1;
                                    end if;
                                end if;
                            end if;
                            bn := bn+1;
                        end do;
                    end do;
                    ans := eva[1];
                else
                    ans := sl[kukan[remcount-1, k, 2], 0];
                end if;
                sum3 := sum3+ans*x^k;
                if degb < k then
                    degb := k;
                end if;
            end if;
        end do;
        fugo[remcount-1, 0] := sum3;
        fugo[remcount-1, 1] := signum(algsubs(x = shita, sum3));
        fugo[remcount-1, 2] := signum(algsubs(x = ue, sum3));
        che := degb;
    end if;
end do;

#実解数計算
    strmbtm := 0;
    strmtop := 0;
    for i from 0 to remcount-1 do
        if fugo[i, 1] = 0 then
            fugo[i, 1] := fugo[i-1, 1];
        end if;
        if fugo[i, 2] = 0 then
            fugo[i, 2] := fugo[i-1, 2];
        end if;
        if 1 <= i then
            if fugo[i-1, 1] <> fugo[i, 1] then
                strmbtm := strmbtm+1;
```

```
            end if;
            if fugo[i-1, 2] <> fugo[i, 2] then
                strmtop := strmtop+1;
            end if;
        end if;
    end do;

#最終出力
    strmans := abs(strmbtm-strmtop);
    print("correct", [zwc, sn, mey], n);
    print("答え");
    print(strmans);
end proc;
```

## グラハムアルゴリズム 実験用プログラム

```
pointlist := makepointlist(500);
print("====ISCZ 行列計算====");
isczgst := time();
isczggrhmlist := CodeTools[Usage](isczggrhm(pointlist, 2));
isczgpast := time()-isczgst;
print("====ISCZ 式列計算====");
isczsst := time();
isczsgrhmlist := CodeTools[Usage](isczsgrhm(pointlist, 2));
isczspast := time()-isczsst;
print("====厳密計算====");
rnst := time();
rngrhmlist := CodeTools[Usage](rngrhm(pointlist));
rnpast := time()-rnst;
print("====総合====");
print(plot([pointlist, rngrhmlist, isczggrhmlist, isczsgrhmlist],
                                    style = [point, line, line, line]));
print("厳密時間", rnpast);
print("行列時間", isczgpast);
print("式列時間", isczspast);
```

## グラハムアルゴリズム用点集合生成プログラム

| 入力引数 | 内容 |
|---|---|
| num | 生成する点の数 |

```
makepointlist := proc (num)
    local i, pointlist, px, py, c;
    pointlist := {};
    for i to num do
        c := 0;
        while c = 0 do
            px := simplify(sqrt((mod(rand(), 9999)+1)/(mod(rand(), 97)+3)));
            py := simplify(sqrt((mod(rand(), 9999)+1)/(mod(rand(), 97)+3)));
            if is(px^2+py^2 < 100) and is((1/2)*px < py) and is(py < 2*px) then
                pointlist := [op(pointlist), [px, py]];
```

```
                c := 1;
            end if;
        end do;
    end do;
    return pointlist;
end proc;
```

## グラハムアルゴリズム (厳密計算)

| 入力引数 | 内容 |
|---|---|
| pointlist | 点集合 |

```
grhm := proc (pointlist)
    local inputlist, n, m, sortlist, i, j, d, min, slomax, slo, grhmlist,
            Ax, Ay, Bx, By, Cx, Cy, D;

    inputlist := pointlist;
    n := nops(inputlist);
    m := n-1;
    sortlist := {};

#Y 最小点探索
    min := inputlist[1, 2];
    d := 1;
    for i from 2 to n do
        if inputlist[i, 2] < min then
            d := i;
            min := inputlist[i, 2];
        end if;
    end do;
    sortlist := [op(sortlist), inputlist[d]];
    inputlist := subsop(d = NULL, inputlist);

#偏角ソート
    for i to n-1 do
        slomax := (inputlist[1, 1]-sortlist[1, 1])/(inputlist[1, 2]-sortlist[1, 2]);
        d := 1;
        for j from 2 to m do
            slo := (inputlist[j, 1]-sortlist[1, 1])/(inputlist[j, 2]-sortlist[1, 2]);
            if slomax < slo then
                d := j;
                slomax := slo;
            end if;
        end do;
        sortlist := [op(sortlist), inputlist[d]];
        inputlist := subsop(d = NULL, inputlist);
        m := m-1;
    end do;
    sortlist := [op(sortlist), sortlist[1]];
    grhmlist := sortlist;

#凸包構成点探索
```

```
    i := 2;
    while i <= n do
        Ax := grhmlist[i-1, 1];
        Ay := grhmlist[i-1, 2];
        Bx := grhmlist[i, 1];
        By := grhmlist[i, 2];
        Cx := grhmlist[i+1, 1];
        Cy := grhmlist[i+1, 2];
        D := (Bx-Ax)*(Cy-Ay)-(By-Ay)*(Cx-Ax);
##凸包構成点判定
        if D < 0 then
            grhmlist := subsop(i = NULL, grhmlist);
            n := n-1;
            i := i-1;
            if i < 2 then
                i := 2;
            end if;
        else
            i := i+1;
        end if;
    end do;

#最終出力
    plot([sortlist, grhmlist], style = [point, line]);
    return grhmlist;
end proc;
```

## グラハムアルゴリズム (ISCZ 行列法)

| 入力引数 | 内容 |
|---|---|
| pointlist | 点集合 |
| dign | 精度桁 |

```
isczggrhm := proc (pointlist, dign)
    local inputlist, n, m, sortlist, i, j, d, miny, slomax, slo, x, y, xshita, xue,
            yshita, yue, kukansortlist, grhmlist, Axs, Axu, Axn, Ays, Ayu, Ayn,
            Bxs, Bxu, Bxn, Bys, Byu, Byn, Cxs, Cxu, Cxn, Cys, Cyu, Cyn, Ds, Du, Dn,
            Es, Eu, En, Fs, Fu, Fn, Gs, Gu, Gn, Hs, Hu, Hn, Is, Iu, In, Xs, Xu, Xn,
            sl, sn, kukanlist, zwc, mey, kotae, ans, slomaxshi, slomaxbo, sloshi, slobo,
            s, ss, eva, sc, sd, se, yo, pa, pb, eyo, na, nb, sla, slb, plotsortlist;

    inputlist := pointlist;
    n := nops(inputlist);
    m := n-1;
    sortlist := {};
    kukanlist := {};
    kukansortlist := {};
    sn := 1;
    zwc := 0;
    mey := 0;
```

```
#区間化
    for i to n do
        x := inputlist[i, 1];
        xshita := evalf(floor(x*10^dign)/10^dign);
        xue := evalf(ceil(x*10^dign)/10^dign);
        y := inputlist[i, 2];
        yshita := evalf(floor(y*10^dign)/10^dign);
        yue := evalf(ceil(y*10^dign)/10^dign);
        kukanlist := [op(kukanlist), [[xshita, xue, sn], [yshita, yue, sn+1]]];
        sl[sn, 0] := x;
        sl[sn, 1] := -1;
        sl[sn, 2] := -1;
        sl[sn+1, 0] := y;
        sl[sn+1, 1] := -1;
        sl[sn+1, 2] := -1;
        sn := sn+2;
    end do;

#Y 最小点探索
    miny[1] := kukanlist[1, 2, 1];
    miny[2] := kukanlist[1, 2, 2];
    miny[3] := kukanlist[1, 2, 3];
    d := 1;
    for i from 2 to n do
        kotae[1] := miny[1]-kukanlist[i, 2, 2];
        kotae[2] := miny[2]-kukanlist[i, 2, 1];
        kotae[3] := sn;
        sl[sn, 0] := miny[3];
        sl[sn, 1] := kukanlist[i, 2, 3];
        sl[sn, 2] := "-";
        sn := sn+1;
        if is(0 < kotae[2]) then
            d := i;
            miny[1] := kukanlist[i, 2, 1];
            miny[2] := kukanlist[i, 2, 2];
            miny[3] := kukanlist[i, 2, 3];
        else
            if is(kotae[1]*kotae[2] <= 0) then
                zwc := zwc+1;
                ans := inputlist[d, 2]-inputlist[i, 2];
                if is(ans <> 0) then
                    print("minselectNup", [zwc, sn, mey], dign);
                    return isczggrhm(pointlist, dign+1);
                end if;
            end if;
        end if
    end do;
    sortlist := [op(sortlist), inputlist[d]];
    kukansortlist := [op(kukansortlist), kukanlist[d]];
    inputlist := subsop(d = NULL, inputlist);
    kukanlist := subsop(d = NULL, kukanlist);

#偏角ソート
    for i to n-1 do
```

```
slomaxshi[1] := kukanlist[1, 1, 2]-kukansortlist[1, 1, 1];
slomaxshi[2] := kukanlist[1, 1, 1]-kukansortlist[1, 1, 2];
slomaxshi[3] := sn;
sl[sn, 0] := kukanlist[1, 1, 3];
sl[sn, 1] := kukansortlist[1, 1, 3];
sl[sn, 2] := "-";
sn := sn+1;
slomaxbo[1] := kukanlist[1, 2, 2]-kukansortlist[1, 2, 1];
slomaxbo[2] := kukanlist[1, 2, 1]-kukansortlist[1, 2, 2];
slomaxbo[3] := sn;
sl[sn, 0] := kukanlist[1, 2, 3];
sl[sn, 1] := kukansortlist[1, 2, 3];
sl[sn, 2] := "-";
sn := sn+1;
slomax[1] := min(slomaxshi[1]/slomaxbo[1], slomaxshi[1]/slomaxbo[2],
                        slomaxshi[2]/slomaxbo[1], slomaxshi[2]/slomaxbo[2]);
slomax[2] := max(slomaxshi[1]/slomaxbo[1], slomaxshi[1]/slomaxbo[2],
                        slomaxshi[2]/slomaxbo[1], slomaxshi[2]/slomaxbo[2]);
slomax[3] := sn;
sl[sn, 0] := slomaxshi[3];
sl[sn, 1] := slomaxbo[3];
sl[sn, 2] := "÷";
sn := sn+1;
d := 1;
for j from 2 to m do
    sloshi[1] := kukanlist[j, 1, 2]-kukansortlist[1, 1, 1];
    sloshi[2] := kukanlist[j, 1, 1]-kukansortlist[1, 1, 2];
    sloshi[3] := sn;
    sl[sn, 0] := kukanlist[j, 1, 3];
    sl[sn, 1] := kukansortlist[1, 1, 3];
    sl[sn, 2] := "-";
    sn := sn+1;
    slobo[1] := kukanlist[j, 2, 2]-kukansortlist[1, 2, 1];
    slobo[2] := kukanlist[j, 2, 1]-kukansortlist[1, 2, 2];
    slobo[3] := sn;
    sl[sn, 0] := kukanlist[j, 2, 3];
    sl[sn, 1] := kukansortlist[1, 2, 3];
    sl[sn, 2] := "-";
    sn := sn+1;
    slo[1] := min(sloshi[1]/slobo[1], sloshi[1]/slobo[2],
                        sloshi[2]/slobo[1], sloshi[2]/slobo[2]);
    slo[2] := max(sloshi[1]/slobo[1], sloshi[1]/slobo[2],
                        sloshi[2]/slobo[1], sloshi[2]/slobo[2]);
    slo[3] := sn;
    sl[sn, 0] := sloshi[3];
    sl[sn, 1] := slobo[3];
    sl[sn, 2] := "÷";
    sn := sn+1;
    kotae[1] := slo[1]-slomax[2];
    kotae[2] := slo[2]-slomax[1];
    kotae[3] := sn;
    sl[sn, 0] := slo[3];
    sl[sn, 1] := slomax[3];
    sl[sn, 2] := "-";
```

```
            sn := sn+1;
            if is(0 < kotae[1]) then
                d := j;
                slomax[1] := slo[1];
                slomax[2] := slo[2];
                slomax[3] := slo[3];
            else
##ゼロ判定
                if is(kotae[1]*kotae[2] <= 0) then
                    zwc := zwc+1;
                    s := slo[3];
                    ss := s;
                    unassign(eva);
                    eva[0, 1] := sl[s, 0];
                    eva[0, 2] := sl[s, 1];
                    eva[0, 3] := sl[s, 2];
                    eva[0, 0] := 1;
                    sc := 1;
                    sd := 0;
                    se := 0;
                    yo := 1;
###行列展開
                    while 0 < sc do
                        eva[sd+1, 0] := 0;
                        while se < eva[sd, 0] do
                            if eva[sd, 3*se+3] <> -1 then
                                pa := eva[sd, 3*se+1];
                                pb := eva[sd, 3*se+2];
                                eva[sd+1, 3*eva[sd+1, 0]+1] := sl[pa, 0];
                                eva[sd+1, 3*eva[sd+1, 0]+2] := sl[pa, 1];
                                eva[sd+1, 3*eva[sd+1, 0]+3] := sl[pa, 2];
                                eva[sd+1, 3*eva[sd+1, 0]+4] := sl[pb, 0];
                                eva[sd+1, 3*eva[sd+1, 0]+5] := sl[pb, 0];
                                eva[sd+1, 3*eva[sd+1, 0]+6] := sl[pb, 0];
                                eva[sd+1, 0] := eva[sd+1, 0]+2;
                                sc := sc+1;
                                if sl[pa, 2] = -1 then
                                    sc := sc-1;
                                end if;
                                if sl[pb, 2] = -1 then
                                    sc := sc-1;
                                end if;
                            end if;
                            se := se+1;
                        end do;
                        sd := sd+1;
                        yo := yo+eva[sd, 0];
                        se := 0;
                    end do;
                    sc := 0;
                    sd := sd-1;
                    eyo := sd+2+3*yo;
                    if mey < eyo then
                        mey := eyo;
```

```
                end if;
###行列計算
                while 0 <= sd do
                    while se < eva[sd, 0] do
                        if eva[sd, 3*se+3] <> -1 then
                            na := eva[sd+1, 6*sc+1];
                            nb := eva[sd+1, 6*sc+4];
                            sla := eva[sd, 3*se+1];
                            sl[sla, 0] := na;
                            sl[sla, 1] := -1;
                            sl[sla, 2] := -1;
                            slb := eva[sd, 3*se+2];
                            sl[slb, 0] := nb;
                            sl[slb, 1] := -1;
                            sl[slb, 2] := -1;
                            sc := sc+1;
                            if eva[sd, 3*se+3] = "+" then
                                eva[sd, 3*se+1] := simplify(combine(na+nb));
                            end if;
                            if eva[sd, 3*se+3] = "-" then
                                eva[sd, 3*se+1] := simplify(combine(na-nb));
                            end if;
                            if eva[sd, 3*se+3] = "×" then
                                eva[sd, 3*se+1] := simplify(combine(na*nb));
                            end if;
                            if eva[sd, 3*se+3] = "÷" then
                                if nb = 0 then
                                    print("0divNup", [zwc, sn, mey], dign);
                                    return isczggrhm(pointlist, dign+1);
                                end if;
                                eva[sd, 3*se+1] := simplify(combine(na/nb));
                            end if;
                            eva[sd, 3*se+2] := -1;
                            eva[sd, 3*se+3] := -1;
                        end if;
                        se := se+1;
                    end do;
                    sd := sd-1;
                    sc := 0;
                    se := 0;
                end do;

                ans := eva[0, 1];
                if is(ans <> 0) then
                    print("sortNup", [zwc, sn, mey], dign);
                    return isczggrhm(pointlist, dign+1);
                end if;
            end if;
        end if;
    end do;
    sortlist := [op(sortlist), inputlist[d]];
    kukansortlist := [op(kukansortlist), kukanlist[d]];
    inputlist := subsop(d = NULL, inputlist);
    kukanlist := subsop(d = NULL, kukanlist);
```

```
        m := m-1
    end do;
    sortlist := [op(sortlist), sortlist[1]];
    kukansortlist := [op(kukansortlist), kukansortlist[1]];
    plotsortlist := sortlist;
    grhmlist := kukansortlist;

#凸包構成点探索
    i := 2;
    while i < n do
        Axs := grhmlist[i-1, 1, 1];
        Axu := grhmlist[i-1, 1, 2];
        Axn := grhmlist[i-1, 1, 3];
        Ays := grhmlist[i-1, 2, 1];
        Ayu := grhmlist[i-1, 2, 2];
        Ayn := grhmlist[i-1, 2, 3];
        Bxs := grhmlist[i, 1, 1];
        Bxu := grhmlist[i, 1, 2];
        Bxn := grhmlist[i, 1, 3];
        Bys := grhmlist[i, 2, 1];
        Byu := grhmlist[i, 2, 2];
        Byn := grhmlist[i, 2, 3];
        Cxs := grhmlist[i+1, 1, 1];
        Cxu := grhmlist[i+1, 1, 2];
        Cxn := grhmlist[i+1, 1, 3];
        Cys := grhmlist[i+1, 2, 1];
        Cyu := grhmlist[i+1, 2, 2];
        Cyn := grhmlist[i+1, 2, 3];
        Ds := Bxs-Axu;
        Du := Bxu-Axs;
        Dn := sn;
        sl[sn, 0] := Bxn;
        sl[sn, 1] := Axn;
        sl[sn, 2] := "-";
        sn := sn+1;
        Es := Cys-Ayu;
        Eu := Cyu-Ays;
        En := sn;
        sl[sn, 0] := Cyn;
        sl[sn, 1] := Ayn;
        sl[sn, 2] := "-";
        sn := sn+1;
        Fs := Bys-Ayu;
        Fu := Byu-Ays;
        Fn := sn;
        sl[sn, 0] := Byn;
        sl[sn, 1] := Ayn;
        sl[sn, 2] := "-";
        sn := sn+1;
        Gs := Cxs-Axu;
        Gu := Cxu-Axs;
        Gn := sn;
        sl[sn, 0] := Cxn;
        sl[sn, 1] := Axn;
```

```
        sl[sn, 2] := "-";
        sn := sn+1;
        Hs := min(Ds*Es, Ds*Eu, Du*Es, Du*Eu);
        Hu := max(Ds*Es, Ds*Eu, Du*Es, Du*Eu);
        Hn := sn;
        sl[sn, 0] := Dn;
        sl[sn, 1] := En;
        sl[sn, 2] := "×";
        sn := sn+1;
        Is := min(Fs*Gs, Fs*Gu, Fu*Gs, Fu*Gu);
        Iu := max(Fs*Gs, Fs*Gu, Fu*Gs, Fu*Gu);
        In := sn;
        sl[sn, 0] := Fn;
        sl[sn, 1] := Gn;
        sl[sn, 2] := "×";
        sn := sn+1;
        Xs := Hs-Iu;
        Xu := Hu-Is;
        Xn := sn;
        sl[sn, 0] := Hn;
        sl[sn, 1] := In;
        sl[sn, 2] := "-";
        sn := sn+1;
##ゼロ判定
        if Xs*Xu <= 0 then
            zwc := zwc+1;
            s := Xn;
            ss := s;
            unassign(eva);
            eva[0, 1] := sl[s, 0];
            eva[0, 2] := sl[s, 1];
            eva[0, 3] := sl[s, 2];
            eva[0, 0] := 1;
            sc := 1;
            sd := 0;
            se := 0;
            yo := 1;
###行列展開
            while 0 < sc do
                eva[sd+1, 0] := 0;
                while se < eva[sd, 0] do
                    if eva[sd, 3*se+3] <> -1 then
                        pa := eva[sd, 3*se+1];
                        pb := eva[sd, 3*se+2];
                        eva[sd+1, 3*eva[sd+1, 0]+1] := sl[pa, 0];
                        eva[sd+1, 3*eva[sd+1, 0]+2] := sl[pa, 1];
                        eva[sd+1, 3*eva[sd+1, 0]+3] := sl[pa, 2];
                        eva[sd+1, 3*eva[sd+1, 0]+4] := sl[pb, 0];
                        eva[sd+1, 3*eva[sd+1, 0]+5] := sl[pb, 0];
                        eva[sd+1, 3*eva[sd+1, 0]+6] := sl[pb, 0];
                        eva[sd+1, 0] := eva[sd+1, 0]+2;
                        sc := sc+1;
                        if sl[pa, 2] = -1 then
                            sc := sc-1;
```

```
                end if;
                if sl[pb, 2] = -1 then
                    sc := sc-1;
                end if;
            end if;
            se := se+1;
        end do;
        sd := sd+1;
        yo := yo+eva[sd, 0];
        se := 0;
    end do;
    sc := 0;
    sd := sd-1;
    eyo := sd+2+3*yo;
    if mey < eyo then
        mey := eyo;
    end if;
###行列計算
    while 0 <= sd do
        while se < eva[sd, 0] do
            if eva[sd, 3*se+3] <> -1 then
                na := eva[sd+1, 6*sc+1];
                nb := eva[sd+1, 6*sc+4];
                sla := eva[sd, 3*se+1];
                sl[sla, 0] := na;
                sl[sla, 1] := -1;
                sl[sla, 2] := -1;
                slb := eva[sd, 3*se+2];
                sl[slb, 0] := nb;
                sl[slb, 1] := -1;
                sl[slb, 2] := -1;
                sc := sc+1;
                if eva[sd, 3*se+3] = "+" then
                    eva[sd, 3*se+1] := simplify(combine(na+nb));
                end if;
                if eva[sd, 3*se+3] = "-" then
                    eva[sd, 3*se+1] := simplify(combine(na-nb));
                end if;
                if eva[sd, 3*se+3] = "×" then
                    eva[sd, 3*se+1] := simplify(combine(na*nb));
                end if;
                if eva[sd, 3*se+3] = "÷" then
                    if nb = 0 then
                        print("0divNup", [zwc, sn, mey], dign);
                        return isczggrhm(pointlist, dign+1);
                    end if;
                    eva[sd, 3*se+1] := simplify(combine(na/nb));
                end if;
                eva[sd, 3*se+2] := -1;
                eva[sd, 3*se+3] := -1;
            end if;
            se := se+1;
        end do;
        sd := sd-1;
```

```
                sc := 0;
                se := 0;
            end do;

            ans := eva[0, 1];
            if ans = 0 then
                Xs := 0;
                Xu := 0;
                sl[ss, 0] := 0;
                sl[ss, 1] := -1;
                sl[ss, 2] := -1;
            else
                print("Nup", [zwc, sn, mey], dign);
                return isczggrhm(pointlist, dign+1);
            end if;
        end if;

##凸包構成点判定
        if Xu < 0 then
            grhmlist := subsop(i = NULL, grhmlist);
            sortlist := subsop(i = NULL, sortlist);
            n := n-1;
            i := i-1;
            if i < 2 then
                i := 2;
            end if;
        else
            i := i+1;
        end if;
    end do;

#最終出力
    print("collect", [zwc, sn, mey], dign);
    print(plot([plotsortlist, sortlist], style = [point, line]));
    return sortlist;
end proc;
```

## グラハムアルゴリズム (ISCZ 式列法)

| 入力引数 | 内容 |
|---|---|
| pointlist | 点集合 |
| dign | 精度桁 |

```
isczsgrhm := proc (pointlist, dign)
    local inputlist, n, m, sortlist, i, j, d, miny, slomax, slo, x, y, xshita, xue,
            yshita, yue, kukansortlist, grhmlist, Axs, Axu, Axn, Ays, Ayu, Ayn,
            Bxs, Bxu, Bxn, Bys, Byu, Byn, Cxs, Cxu, Cxn, Cys, Cyu, Cyn, Ds, Du, Dn,
            Es, Eu, En, Fs, Fu, Fn, Gs, Gu, Gn, Hs, Hu, Hn, Is, Iu, In, Xs, Xu, Xn,
            sl, sn, kukanlist, zwc, mey, kotae, ans, slomaxshi, slomaxbo, sloshi, slobo,
            s, ss, eva, en, el, mel, bn, e, ean, es, ey, plotsortlist;
```

```
    inputlist := pointlist;
    n := nops(inputlist);
    m := n-1;
    sortlist := {};
    kukanlist := {};
    kukansortlist := {};
    sn := 1;
    zwc := 0;
    mey := 0;

#区間化
    for i to n do
        x := inputlist[i, 1];
        xshita := evalf(floor(x*10^dign)/10^dign);
        xue := evalf(ceil(x*10^dign)/10^dign);
        y := inputlist[i, 2];
        yshita := evalf(floor(y*10^dign)/10^dign);
        yue := evalf(ceil(y*10^dign)/10^dign);
        kukanlist := [op(kukanlist), [[xshita, xue, sn], [yshita, yue, sn+1]]];
        sl[sn, 0] := x;
        sl[sn, 1] := -1;
        sl[sn, 2] := -1;
        sl[sn+1, 0] := y;
        sl[sn+1, 1] := -1;
        sl[sn+1, 2] := -1;
        sn := sn+2
    end do;

#Y 最小点探索
    miny[1] := kukanlist[1, 2, 1];
    miny[2] := kukanlist[1, 2, 2];
    miny[3] := kukanlist[1, 2, 3];
    d := 1;
    for i from 2 to n do
        kotae[1] := miny[1]-kukanlist[i, 2, 2];
        kotae[2] := miny[2]-kukanlist[i, 2, 1];
        kotae[3] := sn;
        sl[sn, 0] := miny[3];
        sl[sn, 1] := kukanlist[i, 2, 3];
        sl[sn, 2] := "-";
        sn := sn+1;
        if is(0 < kotae[2]) then
            d := i;
            miny[1] := kukanlist[i, 2, 1];
            miny[2] := kukanlist[i, 2, 2];
            miny[3] := kukanlist[i, 2, 3];
        else
            if is(kotae[1]*kotae[2] <= 0) then
                zwc := zwc+1;
                ans := inputlist[d, 2]-inputlist[i, 2];
                if is(ans <> 0) then
                    print("minselectNup", [zwc, sn, mey], dign);
                    return isczggrhm(pointlist, dign+1);
                end if;
```

```
            end if;
        end if;
    end do;
    sortlist := [op(sortlist), inputlist[d]];
    kukansortlist := [op(kukansortlist), kukanlist[d]];
    inputlist := subsop(d = NULL, inputlist);
    kukanlist := subsop(d = NULL, kukanlist);

#偏角ソート
    for i to n-1 do
        slomaxshi[1] := kukanlist[1, 1, 2]-kukansortlist[1, 1, 1];
        slomaxshi[2] := kukanlist[1, 1, 1]-kukansortlist[1, 1, 2];
        slomaxshi[3] := sn;
        sl[sn, 0] := kukanlist[1, 1, 3];
        sl[sn, 1] := kukansortlist[1, 1, 3];
        sl[sn, 2] := "-";
        sn := sn+1;
        slomaxbo[1] := kukanlist[1, 2, 2]-kukansortlist[1, 2, 1];
        slomaxbo[2] := kukanlist[1, 2, 1]-kukansortlist[1, 2, 2];
        slomaxbo[3] := sn;
        sl[sn, 0] := kukanlist[1, 2, 3];
        sl[sn, 1] := kukansortlist[1, 2, 3];
        sl[sn, 2] := "-";
        sn := sn+1;
        slomax[1] := min(slomaxshi[1]/slomaxbo[1], slomaxshi[1]/slomaxbo[2],
                         slomaxshi[2]/slomaxbo[1], slomaxshi[2]/slomaxbo[2]);
        slomax[2] := max(slomaxshi[1]/slomaxbo[1], slomaxshi[1]/slomaxbo[2],
                         slomaxshi[2]/slomaxbo[1], slomaxshi[2]/slomaxbo[2]);
        slomax[3] := sn;
        sl[sn, 0] := slomaxshi[3];
        sl[sn, 1] := slomaxbo[3];
        sl[sn, 2] := "÷";
        sn := sn+1;
        d := 1;
        for j from 2 to m do
            sloshi[1] := kukanlist[j, 1, 2]-kukansortlist[1, 1, 1];
            sloshi[2] := kukanlist[j, 1, 1]-kukansortlist[1, 1, 2];
            sloshi[3] := sn;
            sl[sn, 0] := kukanlist[j, 1, 3];
            sl[sn, 1] := kukansortlist[1, 1, 3];
            sl[sn, 2] := "-";
            sn := sn+1;
            slobo[1] := kukanlist[j, 2, 2]-kukansortlist[1, 2, 1];
            slobo[2] := kukanlist[j, 2, 1]-kukansortlist[1, 2, 2];
            slobo[3] := sn;
            sl[sn, 0] := kukanlist[j, 2, 3];
            sl[sn, 1] := kukansortlist[1, 2, 3];
            sl[sn, 2] := "-";
            sn := sn+1;
            slo[1] := min(sloshi[1]/slobo[1], sloshi[1]/slobo[2],
                          sloshi[2]/slobo[1], sloshi[2]/slobo[2]);
            slo[2] := max(sloshi[1]/slobo[1], sloshi[1]/slobo[2],
                          sloshi[2]/slobo[1], sloshi[2]/slobo[2]);
            slo[3] := sn;
```

```
            sl[sn, 0] := sloshi[3];
            sl[sn, 1] := slobo[3];
            sl[sn, 2] := "÷";
            sn := sn+1;
            kotae[1] := slo[1]-slomax[2];
            kotae[2] := slo[2]-slomax[1];
            kotae[3] := sn;
            sl[sn, 0] := slo[3];
            sl[sn, 1] := slomax[3];
            sl[sn, 2] := "-";
            sn := sn+1;
            if is(0 < kotae[1]) then
                d := j;
                slomax[1] := slo[1];
                slomax[2] := slo[2];
                slomax[3] := slo[3];
            else
##ゼロ判定
                if is(kotae[1]*kotae[2] <= 0) then
                    zwc := zwc+1;
                    s := slo[3];
                    ss := s;
                    unassign(eva);
                    eva[0] := sl[s, 2];
                    eva[1] := s;
                    eva[2] := "s";
                    eva[3] := sl[s, 0];
                    eva[4] := "s";
                    eva[5] := sl[s, 1];
                    en := 2;
                    el := 6;
                    mel := 6;
###式列展開・計算
                    while 0 < en do
                        bn := 0;
                        while bn < el do
                            if eva[bn] = "s" then
                                if sl[eva[bn+1], 2] = -1 then
                                    eva[bn] := "r";
                                    eva[bn+1] := sl[eva[bn+1], 0];
                                    en := en-1;
                                    if eva[bn-2] = "+"
                                      or eva[bn-2] = "-"
                                      or eva[bn-2] = "×"
                                      or eva[bn-2] = "÷" then
                                        bn := bn-3;
                                    else
                                        bn := bn-5;
                                    end if;
                                    if bn < 0 then
                                        bn := -1;
                                    end if;
                                else
                                    s := eva[bn+1];
```

```
            for e from el-1 by -1 to bn+2 do
                eva[e+4] := eva[e];
            end do;
            eva[bn] := sl[s, 2];
            eva[bn+1] := s;
            eva[bn+2] := "s";
            eva[bn+3] := sl[s, 0];
            eva[bn+4] := "s";
            eva[bn+5] := sl[s, 1];
            en := en+1;
            el := el+4;
            bn := bn-1;
            if mel < el then
                mel := el;
            end if;
        end if;
    else
        if eva[bn+2] = "r" and eva[bn+4] = "r" then
            if eva[bn] = "+" then
                ean := simplify(combine(eva[bn+3]+eva[bn+5]));
            end if;
            if eva[bn] = "-" then
                ean := simplify(combine(eva[bn+3]-eva[bn+5]));
            end if;
            if eva[bn] = "×" then
                ean := simplify(combine(eva[bn+3]*eva[bn+5]));
            end if;
            if eva[bn] = "÷" then
                if eva[bn+5] = 0 then
                    print("sortNup", [zwc, sn, mel], dign);
                    return isczsgrhm(pointlist, dign+1);
                end if;
                ean := simplify(combine(eva[bn+3]/eva[bn+5]));
            end if;
            es := eva[bn+1];
            if es <> 0 then
                sl[es, 0] := ean;
                sl[es, 1] := -1;
                sl[es, 2] := -1;
                eva[bn] := "r";
                eva[bn+1] := ean;
            end if;
            for ey from bn+2 to el-5 do
                eva[ey] := eva[ey+4];
            end do;
            el := el-4;
            if eva[bn-2] = "+"
              or eva[bn-2] = "-"
              or eva[bn-2] = "×"
              or eva[bn-2] = "÷" then
                bn := bn-3;
            else
                bn := bn-5;
            end if;
```

```
                                if bn < 0 then
                                    bn := -1;
                                end if;
                            end if;
                        end if;
                        bn := bn+1;
                    end do;
                end do;

                ans := eva[1];
                if is(ans <> 0) then
                    print("sortNup", [zwc, sn, mey], dign);
                    return isczggrhm(pointlist, dign+1);
                end if;
            end if;
        end if;
    end do;
    sortlist := [op(sortlist), inputlist[d]];
    kukansortlist := [op(kukansortlist), kukanlist[d]];
    inputlist := subsop(d = NULL, inputlist);
    kukanlist := subsop(d = NULL, kukanlist);
    m := m-1;
end do;
sortlist := [op(sortlist), sortlist[1]];
kukansortlist := [op(kukansortlist), kukansortlist[1]];
plotsortlist := sortlist;
grhmlist := kukansortlist;

#凸包構成点探索
    i := 2;
    while i < n do
        Axs := grhmlist[i-1, 1, 1];
        Axu := grhmlist[i-1, 1, 2];
        Axn := grhmlist[i-1, 1, 3];
        Ays := grhmlist[i-1, 2, 1];
        Ayu := grhmlist[i-1, 2, 2];
        Ayn := grhmlist[i-1, 2, 3];
        Bxs := grhmlist[i, 1, 1];
        Bxu := grhmlist[i, 1, 2];
        Bxn := grhmlist[i, 1, 3];
        Bys := grhmlist[i, 2, 1];
        Byu := grhmlist[i, 2, 2];
        Byn := grhmlist[i, 2, 3];
        Cxs := grhmlist[i+1, 1, 1];
        Cxu := grhmlist[i+1, 1, 2];
        Cxn := grhmlist[i+1, 1, 3];
        Cys := grhmlist[i+1, 2, 1];
        Cyu := grhmlist[i+1, 2, 2];
        Cyn := grhmlist[i+1, 2, 3];
        Ds := Bxs-Axu;
        Du := Bxu-Axs;
        Dn := sn;
        sl[sn, 0] := Bxn;
        sl[sn, 1] := Axn;
```

```
        sl[sn, 2] := "-";
        sn := sn+1;
        Es := Cys-Ayu;
        Eu := Cyu-Ays;
        En := sn;
        sl[sn, 0] := Cyn;
        sl[sn, 1] := Ayn;
        sl[sn, 2] := "-";
        sn := sn+1;
        Fs := Bys-Ayu;
        Fu := Byu-Ays;
        Fn := sn;
        sl[sn, 0] := Byn;
        sl[sn, 1] := Ayn;
        sl[sn, 2] := "-";
        sn := sn+1;
        Gs := Cxs-Axu;
        Gu := Cxu-Axs;
        Gn := sn;
        sl[sn, 0] := Cxn;
        sl[sn, 1] := Axn;
        sl[sn, 2] := "-";
        sn := sn+1;
        Hs := min(Ds*Es, Ds*Eu, Du*Es, Du*Eu);
        Hu := max(Ds*Es, Ds*Eu, Du*Es, Du*Eu);
        Hn := sn;
        sl[sn, 0] := Dn;
        sl[sn, 1] := En;
        sl[sn, 2] := "×";
        sn := sn+1;
        Is := min(Fs*Gs, Fs*Gu, Fu*Gs, Fu*Gu);
        Iu := max(Fs*Gs, Fs*Gu, Fu*Gs, Fu*Gu);
        In := sn;
        sl[sn, 0] := Fn;
        sl[sn, 1] := Gn;
        sl[sn, 2] := "×";
        sn := sn+1;
        Xs := Hs-Iu;
        Xu := Hu-Is;
        Xn := sn;
        sl[sn, 0] := Hn;
        sl[sn, 1] := In;
        sl[sn, 2] := "-";
        sn := sn+1;
##ゼロ判定
        if Xs*Xu <= 0 then
            zwc := zwc+1;
            s := Xn;
            ss := s;
            unassign(eva);
            eva[0] := sl[s, 2];
            eva[1] := s;
            eva[2] := "s";
            eva[3] := sl[s, 0];
```

```
            eva[4] := "s";
            eva[5] := sl[s, 1];
            en := 2;
            el := 6;
            mel := 6;
###式列展開・計算
            while 0 < en do
                bn := 0;
                while bn < el do
                    if eva[bn] = "s" then
                        if sl[eva[bn+1], 2] = -1 then
                            eva[bn] := "r";
                            eva[bn+1] := sl[eva[bn+1], 0];
                            en := en-1;
                            if eva[bn-2] = "+"
                              or eva[bn-2] = "-"
                              or eva[bn-2] = "×"
                              or eva[bn-2] = "÷" then
                                bn := bn-3;
                            else
                                bn := bn-5;
                            end if;
                            if bn < 0 then
                                bn := -1;
                            end if;
                        else
                            s := eva[bn+1];
                            for e from el-1 by -1 to bn+2 do
                                eva[e+4] := eva[e];
                            end do;
                            eva[bn] := sl[s, 2];
                            eva[bn+1] := s;
                            eva[bn+2] := "s";
                            eva[bn+3] := sl[s, 0];
                            eva[bn+4] := "s";
                            eva[bn+5] := sl[s, 1];
                            en := en+1;
                            el := el+4;
                            bn := bn-1;
                            if mel < el then
                                mel := el;
                            end if;
                        end if;
                    else
                        if eva[bn+2] = "r" and eva[bn+4] = "r" then
                            if eva[bn] = "+" then
                                ean := simplify(combine(eva[bn+3]+eva[bn+5]));
                            end if;
                            if eva[bn] = "-" then
                                ean := simplify(combine(eva[bn+3]-eva[bn+5]));
                            end if;
                            if eva[bn] = "×" then
                                ean := simplify(combine(eva[bn+3]*eva[bn+5]));
                            end if;
```

```
                            if eva[bn] = "÷" then
                                if eva[bn+5] = 0 then
                                    print("0divNup", [zwc, sn, mel], dign);
                                    return isczsgrhm(pointlist, dign+1);
                                end if;
                                ean := simplify(combine(eva[bn+3]/eva[bn+5]));
                            end if;
                            es := eva[bn+1];
                            if es <> 0 then
                                sl[es, 0] := ean;
                                sl[es, 1] := -1;
                                sl[es, 2] := -1;
                                eva[bn] := "r";
                                eva[bn+1] := ean;
                            end if;
                            for ey from bn+2 to el-5 do
                                eva[ey] := eva[ey+4];
                            end do;
                            el := el-4;
                            if eva[bn-2] = "+"
                              or eva[bn-2] = "-"
                              or eva[bn-2] = "×"
                              or eva[bn-2] = "÷" then
                                bn := bn-3;
                            else
                                bn := bn-5;
                            end if;
                            if bn < 0 then
                                bn := -1;
                            end if;
                        end if;
                    end if;
                    bn := bn+1;
                end do;
            end do;

            ans := eva[1];
            if ans = 0 then
                Xs := 0;
                Xu := 0;
                sl[ss, 0] := 0;
                sl[ss, 1] := -1;
                sl[ss, 2] := -1;
            else
                print("Nup", evalf(ans), [zwc, sn, mel], dign);
                return isczsgrhm(pointlist, dign+1);
            end if;
        end if;

##凸包構成点判定
        if Xu < 0 then
            grhmlist := subsop(i = NULL, grhmlist);
            sortlist := subsop(i = NULL, sortlist);
            n := n-1;
```

```
            i := i-1;
            if i < 2 then
                i := 2;
            end if;
        else
            i := i+1;
        end if;
    end do;

#最終出力
    print("collect", [zwc, sn, mel], dign);
    print(plot([plotsortlist, sortlist], style = [point, line]));
    return sortlist;
end proc;
```

# 付録 2:JavaScript プログラムソース

今回は全ての変数をグローバル変数としている。

## スツルムアルゴリズム入出力関数

```
function iscz_strm(){
    std = new Date();
    lim=7;
    ket = parseFloat(document.F1.keta.value);
//入力
    htop = parseFloat(document.F1.htop.value);
    hbtm = parseFloat(document.F1.hbtm.value);

    up = new Array(lim+1);
    up[7] = parseFloat(document.F1.up7.value);
    up[6] = parseFloat(document.F1.up6.value);
    up[5] = parseFloat(document.F1.up5.value);
    up[4] = parseFloat(document.F1.up4.value);
    up[3] = parseFloat(document.F1.up3.value);
    up[2] = parseFloat(document.F1.up2.value);
    up[1] = parseFloat(document.F1.up1.value);
    up[0] = parseFloat(document.F1.up0.value);

    bt = new Array(lim+1);
    bt[7] = parseFloat(document.F1.bt7.value);
    bt[6] = parseFloat(document.F1.bt6.value);
    bt[5] = parseFloat(document.F1.bt5.value);
    bt[4] = parseFloat(document.F1.bt4.value);
    bt[3] = parseFloat(document.F1.bt3.value);
    bt[2] = parseFloat(document.F1.bt2.value);
    bt[1] = parseFloat(document.F1.bt1.value);
    bt[0] = parseFloat(document.F1.bt0.value);

    stketa = ket;
    edketa = iscz_strm_main(ket);

    document.write("===Keta===<br>");
    document.write("start/" + stketa + "<br>");
    document.write("finish/" + edketa + "<br>");

//経過時間計算
    edd = new Date();
    sth = std.getHours();
    stm = std.getMinutes();
    sts = std.getSeconds();
    stms = std.getMilliseconds();

    edh = edd.getHours();
    edm = edd.getMinutes();
    eds = edd.getSeconds();
    edms = edd.getMilliseconds();
```

```
    elh = edh - sth;
    elm = edm - stm;
    if (elm<0){
        elm=elm+60;
        elh--;
    }
    els = eds - sts;
    if (els<0){
        els=els+60;
        elm--;
    }
    elms = edms - stms;
    if (elms<0){
        elms=elms+1000;
        els--;
    }
    document.write("===Time===<br>");
    document.write("start/" + sth + ":" + stm + ":" + sts + ":" + stms + "<br>");
    document.write("finish/" + edh + ":" + edm + ":" + eds + ":" + edms + "<br>");
    document.write("elapse/" + elh + ":" + elm + ":" + els + ":" + elms + "<br>");
}
```

## スツルムアルゴリズム (ISCZ 式列法)

| 入力引数 | 内容 |
|---|---|
| keta | 精度桁 |

```
function iscz_strm_main(keta){
    document.write("KETA||"+keta+"<br>");

    strm = new Array();

    for(i=lim; i>=0; i--){
        strm[i] = new Array();
        for(j=lim; j>=0; j--){
            strm[i][j] = new Array(3);
        }
    }

//シンボルリスト
//sl[保存位置][シンボル番号]
    sl = new Array(3);

    for(i=0; i<3; i++){
        sl[i] = new Array();
    }
    sl[0][0]=-1;
    sl[1][0]=1;
    sl[2][0]="r";
    sn=1;
```

```
    deg = new Array();
    cosum = new Array();

//入力の出力
    document.write("INPUT F(x)||");
    tpd=-1;
    for (op=lim; op>=0; op--){
        if (up[op]!=0){
            if (tpd>=0){
                document.write("+");
            }
            document.write("("+up[op]+"/"+bt[op]+")");
            if (op!=0){
                document.write("x^"+op);
            }
            if (op>=tpd){
                tpd=op;
            }
        }
    }
    document.write("<br>");
    document.write("KUKAN||"+hbtm+"~"+htop+"<br>");

//区間化
    cosum[0]=0;
    for(kki=tpd; kki>=0 ; kki--){
        num = up[kki]/bt[kki];
        num = Math.round(num * Math.pow(10,keta));
        sl[0][sn]=up[kki];
        sl[1][sn]=bt[kki];
        sl[2][sn]="r";
        if (num==0){
            strm[0][kki][0]= 0;
            strm[0][kki][1]= 0;
            strm[0][kki][2]= sn;
            sn++;
        }else{
            strm[0][kki][0]= (num - 1) / Math.pow(10,keta);
            strm[0][kki][1]= (num + 1) / Math.pow(10,keta);
            strm[0][kki][2]= sn;
            sn++;
            cosum[0]+=strm[0][kki][1];
            document.write("[[" + strm[0][kki][0] + " , " + strm[0][kki][1] + "],"
                                                    + strm[0][kki][2] + "]x^" + kki) ;
        }
    }
    document.write("<br>");
    deg[0]=tpd+1;

//微分
    for (bki=tpd; bki>0; bki--){
        num = (bki*up[bki])/bt[bki];
        num = Math.round(num * Math.pow(10,keta));
        sl[0][sn]=bki*up[bki];
```

```
        sl[1][sn]=bt[bki];
        sl[2][sn]="r";
        if (num==0){
            strm[1][bki-1][0]= 0;
            strm[1][bki-1][1]= 0;
            strm[1][bki-1][2]= sn;
            sn++;
        }else{
            strm[1][bki-1][0]= (num - 1) / Math.pow(10,keta);
            strm[1][bki-1][1]= (num + 1) / Math.pow(10,keta);
            strm[1][bki-1][2]= sn;
            sn++;
            cosum[1]+=strm[1][bki-1][1];
            document.write("[[" + strm[1][bki-1][0] + " , " + strm[1][bki-1][1] + "],"
                                                    + strm[1][bki-1][2] + "]x^" + (bki-1) ) ;
        }
    }
    document.write("<br>");
    deg[1]=tpd;

//ゆーくりっど
    eui=0;
    while (cosum[eui+1]!=0 && deg[0]>eui){
        dega=deg[eui];
        degb=deg[eui+1];
        eunum=dega-degb;

        euca=new Array(dega);
        for (euj=0; euj<dega; euj++){
            euca[euj]=new Array(3);
            euca[euj][0]=strm[eui][euj][0];
            euca[euj][1]=strm[eui][euj][1];
            euca[euj][2]=strm[eui][euj][2];
        }

        eucb=new Array(degb);
        for (euk=0; euk<degb; euk++){
            eucb[euk]=new Array(3);
            eucb[euk][0]=strm[eui+1][euk][0];
            eucb[euk][1]=strm[eui+1][euk][1];
            eucb[euk][2]=strm[eui+1][euk][2];
        }

        for (eul=eunum; eul>=0; eul--){
            eucs = new Array(3);
            eucab=euca[degb-1+eul][0];
            eucat=euca[degb-1+eul][1];
            eucbb=eucb[degb-1][0];
            eucbt=eucb[degb-1][1];
            eucs[0]=Math.min((eucab/eucbb), (eucab/eucbt), (eucat/eucbb), (eucat/eucbt));
            eucs[1]=Math.max((eucab/eucbb), (eucab/eucbt), (eucat/eucbb), (eucat/eucbt));
            eucs[2]=sn;
            sl[0][sn]=euca[degb-1+eul][2];
            sl[1][sn]=eucb[degb-1][2];
```

```
    sl[2][sn]="d";
    sn++;
    eucx=new Array(dega);
    for (eum=0; eum<degb; eum++){
        eucx[eum]=new Array(3);
        eucx[eum][0]=Math.min((eucs[0]*eucb[eum][0]), (eucs[0]*eucb[eum][1]),
                                    (eucs[1]*eucb[eum][0]), (eucs[1]*eucb[eum][1]));
        eucx[eum][1]=Math.max((eucs[0]*eucb[eum][0]), (eucs[0]*eucb[eum][1]),
                                    (eucs[1]*eucb[eum][0]), (eucs[1]*eucb[eum][1]));
        eucx[eum][2]=sn;
        sl[0][sn]=eucs[2];
        sl[1][sn]=eucb[eum][2];
        sl[2][sn]="x";
        sn++;
        euca[eum+eul][0]=euca[eum+eul][0]-eucx[eum][1];
        euca[eum+eul][1]=euca[eum+eul][1]-eucx[eum][0];
        sl[0][sn]=euca[eum+eul][2];
        sl[1][sn]=eucx[eum][2];
        sl[2][sn]="m";
        euca[eum+eul][2]=sn;
        sn++;
        if (euca[eum+eul][0]*euca[eum+eul][1]<=0){
            evans=iscz_eva(euca[eum+eul][2]);
            if (evans==0){
                euca[eum+eul][0]=0;
                euca[eum+eul][1]=0;
            }else{
                return iscz_strm_main(keta+1);
            }
        }
    }
}

if(degb==1 && euca[eum+eul][0]==0){
    deg[eui+2]=0;
    cosum[eui+2]=0;
}else{
    deg[eui+2]=-1;
    cosum[eui+2]=0;
    for (eun=0; eun<degb; eun++){
        strm[eui+2][eun][0]=(-1)*euca[eun][1];
        strm[eui+2][eun][1]=(-1)*euca[eun][0];
        strm[eui+2][eun][2]=sn;
        sl[0][sn]=euca[eun][2];
        sl[1][sn]=0;
        sl[2][sn]="x";
        sn++;
        cosum[eui+2]+=Math.abs(strm[eui+2][eun][1]);
        if (euca[eun][0]!=0 && deg[eui+2]<eun){
            deg[eui+2]=eun;
        }
    }
}
deg[eui+2]++;
```

```
        eui++;
    }


//すつるむ
    stbt=new Array(eui);
    sttp=new Array(eui);
    document.write("BTM:"+hbtm+"/TOP:"+htop+"<br>");
    for (sti=0; sti<=eui; sti++){
        sumbt=[0,0];
        sumtp=[0,0];
        stdeg=deg[sti];
        for (stj=0; stj<stdeg; stj++){
            pbt=Math.pow(hbtm,stj);
            ptp=Math.pow(htop,stj);
            sbt=strm[sti][stj][0];
            stp=strm[sti][stj][1];
            sumbt[0]=sumbt[0]+Math.min(sbt*pbt,stp*pbt);
            sumbt[1]=sumbt[1]+Math.max(sbt*pbt,stp*pbt);
            if (sumbt[0]*sumbt[1]<=0){
                sumbt[0]=0;
                sumbt[1]=0;
            }
            sumtp[0]=sumtp[0]+Math.min(sbt*ptp,stp*ptp);
            sumtp[1]=sumtp[1]+Math.max(sbt*ptp,stp*ptp);
            if (sumtp[0]*sumtp[1]<=0){
                sumtp[0]=0;
                sumtp[1]=0;
            }
        }
        if (sumbt[1]<0){
            stbt[sti]=-1;
        }else if (sumbt[1]==0){
            stbt[sti]=0;
        }else{
            stbt[sti]=1;
        }
        if (sumtp[1]<0){
            sttp[sti]=-1;
        }else if (sumtp[1]==0){
            sttp[sti]=0;
        }else{
            sttp[sti]=1;
        }
    }

    coubt=0;
    coutp=0;
    for (cou=1; cou<=eui; cou++){
        if (stbt[cou]==0){
            stbt[cou]=stbt[cou-1];
        }
        if (sttp[cou]==0){
            sttp[cou]=sttp[cou-1];
```

```
        }
        if (stbt[cou]!=stbt[cou-1]){
            coubt++;
        }
        if (sttp[cou]!=sttp[cou-1]){
            coutp++;
        }
    }
    ans=Math.abs(coubt-coutp);
    document.write("Answer/"+ans+"<br>");

    return keta;
}
```

## スツルムアルゴリズム (シンボル評価)

| 入力引数 | 内容 |
|---|---|
| evasn | 評価シンボル番号 |

```
function iscz_eva(evasn){
    if (sl[2][evasn]=="r"){
        return sl[0][evasn]/sl[1][evasn];
    }

    eva=new Array();
    eva[0]=sl[2][evasn];
    eva[1]=evasn;
    eva[2]-"";
    eva[3]="s";
    eva[4]=sl[0][evasn];
    eva[5]="";
    eva[6]="s";
    eva[7]=sl[1][evasn];
    eva[8]="";
    evan=2;
    eval=9;
    mevl=9;
    while(evan>0){
        miln=3;
        while(miln<eval && eval!=3){
            kkmn=eva[miln+1];
            if (eva[miln]=="s"){
                if (sl[2][kkmn]=="r"){
                    eva[miln]="r";
                    eva[miln+1]=sl[0][kkmn];
                    eva[miln+2]=sl[1][kkmn];
                    evan--;
                    if (eva[miln-3]=="r"){
                        miln=miln-9;
                    }else{
                        miln=miln-6;
                    }
```

```
            if (miln<0){
                miln=-3;
            }
        }else{
            for (ido=eval-1; ido>miln+2; ido--){
                eva[ido+6]=eva[ido];
            }
            eva[miln]=sl[2][kkmn];
            eva[miln+1]=kkmn;
            eva[miln+2]="";
            eva[miln+3]="s";
            eva[miln+4]=sl[0][kkmn];
            eva[miln+5]="";
            eva[miln+6]="s";
            eva[miln+7]=sl[1][kkmn];
            eva[miln+8]="";
            evan++;
            eval=eval+6;
            miln=miln-3;
            if (eval>mevl){
                mevl=eval;
            }
        }
    }else{
        if (eva[miln+3]=="r" && eva[miln+6]=="r"){
            if (eva[miln]=="p"){
                evtp=(eva[miln+4]*eva[miln+8])+(eva[miln+7]*eva[miln+5]);
                evbt=eva[miln+5]*eva[miln+8];
            }
            if (eva[miln]=="m"){
                evtp=(eva[miln+4]*eva[miln+8])-(eva[miln+7]*eva[miln+5]);
                evbt=eva[miln+5]*eva[miln+8];
            }
            if (eva[miln]=="x"){
                evtp=eva[miln+4]*eva[miln+7];
                evbt=eva[miln+5]*eva[miln+8];
            }
            if (eva[miln]=="d"){
                if (eva[miln+7]==0){
                    return -1;
                }
                evtp=eva[miln+4]*eva[miln+8];
                evbt=eva[miln+5]*eva[miln+7];
            }
            divn=euclid(evtp, evbt);
            evtp=evtp/divn;
            evbt=evbt/divn;
            if (Math.abs(evtp/evbt)<0){
                evtp=0;
                evbt=1;
            }
            sl[0][kkmn]=evtp;
            sl[1][kkmn]=evbt;
            sl[2][kkmn]="r";
```

```
                    eva[miln]="r";
                    eva[miln+1]=evtp;
                    eva[miln+2]=evbt;
                    for (ido=miln+3; ido<eval-6; ido++){
                        eva[ido]=eva[ido+6];
                    }
                    eval=eval-6;
                    if (eva[miln-3]=="r"){
                        miln=miln-9;
                    }else{
                        miln=miln-6;
                    }
                    if (miln<0){
                        miln=-3;
                    }
                }
            }
            miln=miln+3;
        }
    }

    if (Math.abs(eva[1]/eva[2])<0){
        eva[1]=0;
        eva[2]=1;
    }
    sl[0][evasn]=eva[1];
    sl[1][evasn]=eva[2];
    sl[2][evasn]="r";
    return eva[1]/eva[2];
}
```

## スツルムアルゴリズム (ユークリッドの互除法)

| 入力引数 | 内容 |
|---|---|
| euca | 整数 A |
| eucb | 整数 B |

```
function euclid(euca, eucb){
    if (euca<0){
        euca=euca*(-1);
    }
    if (eucb<0){
        eucb=eucb*(-1);
    }
    if (euca<eucb){
        stoc=euca;
        euca=eucb;
        eucb=stoc;
    }

    while (eucb>0){
```

```
        rem=euca%eucb;
        euca=eucb;
        eucb=rem;
    }

    return euca;
}
```

# 付録 3:HTML ソース

JavaScript での実行の際に用いた、多項式を入力するためのページの HTML ソース

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
<title>JS-ISCZ 法すつるむ</title>
<script type="text/javascript" src="iscz_strm.js">
</script>
</head>
<body>
<form name="F1" action="#">
精度桁:<input type="text" name="keta" value="1" size=3>
下限:<input type="text" name="hbtm" value="-10" size=3>
上限:<input type="text" name="htop" value="10" size=3>
<br>
<input type="text" name="up7" value="0" size=3>/
    <input type="text" name="bt7" value="1" size=3>x<sup>7</sup>+<br>
<input type="text" name="up6" value="0" size=3>/
    <input type="text" name="bt6" value="1" size=3>x<sup>6</sup>+<br>
<input type="text" name="up5" value="0" size=3>/
    <input type="text" name="bt5" value="1" size=3>x<sup>5</sup>+<br>
<input type="text" name="up4" value="0" size=3>/
    <input type="text" name="bt4" value="1" size=3>x<sup>4</sup>+<br>
<input type="text" name="up3" value="0" size=3>/
    <input type="text" name="bt3" value="1" size=3>x<sup>3</sup>+<br>
<input type="text" name="up2" value="0" size=3>/
    <input type="text" name="bt2" value="1" size=3>x<sup>2</sup>+<br>
<input type="text" name="up1" value="0" size=3>/
    <input type="text" name="bt1" value="1" size=3>x+<br>
<input type="text" name="up0" value="0" size=3>/
    <input type="text" name="bt0" value="1" size=3><br>
<br>
<input type="button" value="判定する" onclick="iscz_strm()"><br>
<br>

</form>
</body>
</html>
```