

## 三角関数の三項漸化式による 傾斜楕円の高速生成法

沼田 宗敏<sup>†1</sup> 奥水 大和<sup>†2</sup> 秦野 甯世<sup>†2</sup>  
神谷 和秀<sup>†3</sup> 野村 俊<sup>†3</sup> 二宮 市三<sup>†4</sup>

従来、直線、円、楕円などの幾何学図形の生成法が多数報告され、傾斜楕円についても若干の研究業績がある。本論文では、1点あたり2回の乗算で傾斜楕円を生成する画期的な三項漸化算法を提案する。三角関数の間に成り立つ一対の漸化式が同型であることが、その秘訣である。初期設定や楕円弧描画は従来法と比較して簡単であり、それを従来の高速生成法と同等以上の計算速度で実現する。また、実験によって新算法の有効性を検証する。

### Fast Algorithm for Generating Oblique Ellipse by Using Three-term-recurrence of Trigonometric Functions

MUNETOSHI NUMADA,<sup>†1</sup> HIROYASU KOSHIMIZU,<sup>†2</sup> YASUYO HATANO,<sup>†2</sup>  
KAZUhide KAMIYA,<sup>†3</sup> TAKASHI NOMURA<sup>†3</sup> and ICHIZO NINOMIYA<sup>†4</sup>

Although many useful methods for generating lines, circles, ellipses including oblique ellipses, etc. have been reported so far, there still remain more problems to be solved. Thus this paper proposes an epoch-making three-term recurrence algorithm for generating oblique ellipses with only two multiplications per point. The key of the method consists in the fact that a pair of the three-term recurrences of trigonometric functions are of the same form. Since the proposed method is simpler than the traditional methods both in setting initial conditions and in generating ellipse arc, this method can be superior or equivalent with respect also to computation cost. Numerical experiments have shown its high quality both in speed and accuracy.

#### 1. はじめに

CAD, CG, プロッタ, 数値制御 (NC) などの分野では、直線、円、楕円など基本的な図形の一部または全体を高速・高精度に発生させなければならない。これらの図形の中で楕円、特に傾きを持つ傾斜楕円の生成手法は、円や直線に比較して複雑である。その生成手法を大きく分けると、式 (1) のように判別関数  $f(x, y) = 0$  を用いるノンパラメトリック方式と、式 (2) のように角度パラメータ  $\theta$  を用いるパラメ

トリック方式とがある。

$$f(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F \quad (1)$$

$$\left. \begin{aligned} x &= x_c + a \cos \phi \cos \theta - b \sin \phi \sin \theta \\ y &= y_c + a \sin \phi \cos \theta + b \cos \phi \sin \theta. \end{aligned} \right\} \quad (2)$$

ノンパラメトリック方式には、軸方向における真の楕円と候補ドットとの距離を最小にする軸方向距離最小法<sup>1), 2)</sup>、2つの候補ドットの変位の絶対値  $|f(x, y)|$  を比較して  $|f(x, y)|$  が小さい方の候補ドットを選択する改良型変異比較法<sup>3)</sup>、2候補点の変位の和の符号によって  $|f(x, y)|$  の小さい候補ドットを選択する剰余最小法<sup>4) - 6)</sup> などがある。一般にノンパラメトリック方式は、判別関数の値を最小にするドットを順次選択するため図形描画精度に優れ、また発生ドットを整数で扱うため、グラフィックスディスプレイへの応用に適している。

一方、角度パラメータ  $\theta$  を用いるパラメトリック方式では、まず粗い点列を発生させ、この点間を図形描画装置の最小単位の増分で補間を行う。楕円を構成す

†1 株式会社ロゼフテクノロジー

Lossev Technology Corporation

†2 中京大学情報理工学部

School of Information Science and Technology, Chukyo University

†3 富山県立大学工学部

Faculty of Engineering, Toyama Prefectural University

†4 名古屋大学名誉教授

Professor Emeritus, Nagoya University

る点列を実数で取り扱うため、プロッタや数値制御に適している。この方式では、 $\sin \theta$  や  $\cos \theta$  など三角関数の記述を省くことが効率化の鍵となっている<sup>7)</sup>。三角関数の計算を省くため、加法定理を用いて  $\sin \theta$  と  $\cos \theta$  の計算を行うことができるが、それでも 10 回の乗算と 6 回の加減算が必要である。これに対し、Smith は 4 回の乗算と 4 回の加減算で楕円上の 1 点を計算する手法を提案した<sup>8)</sup>。

ところで、乗算に要する計算量すなわち計算コストは加減算に比べて大きいので、実用上は乗算の計算回数だけで計算コストを比較してもかまわない。これより、Smith の手法は加法定理を用いた手法よりも 2 倍以上高速な優れた手法であることが分かる<sup>9),10)</sup>。しかしながら、Smith の手法には初期化条件が複雑で、プログラムのステップ数が大きいという問題点がある。これに対し、Hearn ら<sup>11)</sup> や Rogers ら<sup>12)</sup> は初期化条件を簡略化する手法を提案した。ところが、これらの手法の 1 ループの乗算回数は Smith の手法に比べて増えてしまった。このため計算コストは Smith の手法よりも逆に大きくなっている。

このような問題を解決するため輿水ら<sup>13)</sup> は、Smith の手法よりも初期化条件が簡単で、かつ計算コストも小さな、連立漸化式を用いた傾斜楕円生成法を提案した。対となる漸化式は一對ごとに 2 点を生成でき、これに要する乗算回数は 4 回である。このため、楕円上の 1 点を計算する乗算回数はわずか 2 回で、Smith の手法に比べて 1/2 となっている。しかし、これらの 2 点間の角度パラメータ  $\theta$  の位相は互いに  $\pi/2 - \Delta\theta/2$  だけずれており、取扱いがとても難しい。このため、楕円弧生成の計算コストは Smith の手法と大差ない。これより、初期化条件が簡単なだけでなく、楕円弧描画も簡単な傾斜楕円の生成法が求められている。

そこで本論文では、楕円をパラメトリック表現する立場から、三角関数の間の三項漸化関係を利用した傾斜楕円の効率的な計算方法を提案する。これは Smith の手法に比べ、計算コストが 1/2 で初期化条件の計算も簡単なだけでなく、角度パラメータ  $\theta$  の増加にともなって楕円上の軌跡を順次描くことができる。このため、一對の漸化式で発生した 2 点は互いに離れることがなく、輿水的手法に比べて楕円弧生成の計算コストが 1/2 になる。

本論文の構成は以下のとおりである。2 章では従来の傾斜楕円の高速発生手法について述べる。3 章では三項漸化式による傾斜楕円の発生手法を提案する。4 章で効果について検証する。そして、5 章で全体をまとめる。

## 2. 従来の傾斜楕円の高速発生法

本章では、従来の傾斜楕円の高速発生法について概説する。

### 2.1 加法定理を用いた計算法

まず、傾斜楕円の中心を  $(x_c, y_c)$ 、互いに直交する 2 本の主軸および副軸の半径を  $a, b$ 、傾きを  $\phi$  とし、 $A = \cos \phi, B = \sin \phi$  とおく。また、楕円を構成する点列数を  $N$ 、刻み角を  $\Delta\theta = 2\pi/N$ 、角度パラメータを  $\theta_n = n \Delta\theta$  とする。式 (2) の  $\cos \theta_n, \sin \theta_n$  は、あらかじめ  $\cos \Delta\theta, \sin \Delta\theta$  を計算しておくことにより、加法定理から順次計算することができる。これより、 $x'_n = a \cos \theta_n, y'_n = b \sin \theta_n$  において次式を得る。

$$\left. \begin{aligned} x_n &= x_c + x'_n A - y'_n B \\ y_n &= y_c + x'_n B + y'_n A. \end{aligned} \right\} \quad (3)$$

この計算は、初期値を  $\theta_0 = 0$  として、式 (3) の計算を  $N$  回繰り返せばよい。1 点を生成するための乗算回数は、加法定理による  $\cos \theta_n, \sin \theta_n$  の計算に 4 回、 $x'_n, y'_n$  の計算に 2 回、式 (3) の計算に 4 回の計 10 回である<sup>8)</sup>。

### 2.2 Smith の手法

Smith の手法による楕円生成法を図 1 に示す。点列の構成する多角形の面積は、同一点数で構成される楕円の内接多角形の中では最大となるので<sup>8)</sup>、最も効率的な楕円の表現手法である。計算手順は以下のとおりである。

[ Initialize ]

$$\Delta\theta = 2\pi/N$$

$$CT = \cos \phi; ST = \sin \phi$$

$$CDP = \cos \Delta\theta; SDP = \sin \Delta\theta$$

$$A = CDP + SDP \cdot ST \cdot CT (a/b - b/a)$$

$$B = -SDP \{ (b \cdot ST)^2 + (a \cdot CT)^2 \} / ab$$

$$C = SDP \{ (b \cdot CT)^2 + (a \cdot ST)^2 \} / ab$$

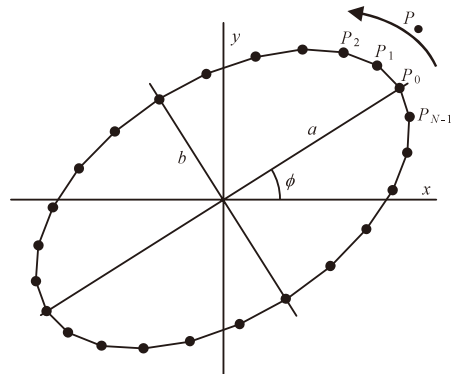


図 1 Smith の楕円生成法

Fig. 1 Smith's ellipse generating method.

$$D = CDP + SDP \cdot ST \cdot CT(b/a - a/b)$$

$$D = D - (C \cdot B)/A$$

$$C = C/A$$

$$x_0 = a \cdot CT; y_0 = a \cdot ST$$

[ Loop ]  $n = 0, 1, \dots, N - 1$

$$\text{plot}(x_c + x_n, y_c + y_n)$$

$$x_{n+1} = A \cdot x_n + B \cdot y_n$$

$$y_{n+1} = C \cdot x_{n+1} + D \cdot y_n.$$

1点を生成するための乗算回数は4回で、加法定理を用いた手法に比べ計算コストは半分である。

2.3 輿水の手法

輿水による傾斜楕円発生法を以下に示す。

まず、式(2)の傾斜楕円で中心を(0, 0)とおく。

$$\left. \begin{aligned} x_n &= a \cos \phi \cos \theta_n - b \sin \phi \sin \theta_n \\ y_n &= a \sin \phi \cos \theta_n + b \cos \phi \sin \theta_n. \end{aligned} \right\} (4)$$

ここで、 $x_n$ とパラメータ $\theta_n$ の位相が $-\pi/2 + \Delta\theta/2$ 分だけずれた数列 $x'_n$ を、次式のように定める。

$$\left. \begin{aligned} x'_n &= a \cos \phi \sin(\theta_n + \Delta\theta/2) \\ &+ b \sin \phi \cos(\theta_n + \Delta\theta/2). \end{aligned} \right\} (5)$$

これより、 $r_x^2 = a^2 \cos^2 \phi + b^2 \sin^2 \phi$ 、 $\tan \xi_x = (b/a) \tan \phi$ とおけば、数列 $x_n$ と $x'_n$ は次式で表現できる。

$$\left. \begin{aligned} x_n &= r_x \cos(\xi_x + \theta_n) \\ x'_n &= r_x \sin(\xi_x + \theta_n + \Delta\theta/2) \end{aligned} \right\} (6)$$

上式は、 $\varepsilon = 2 \sin(\Delta\theta/2)$ とおくことにより、次式の連立漸化式で計算できる<sup>7),14)</sup>。

$$\left. \begin{aligned} x_{n+1} &= x_n - \varepsilon \cdot x'_n \\ x'_{n+1} &= x'_n + \varepsilon \cdot x_{n+1}. \end{aligned} \right\} (7)$$

$y_n$ も同様にして計算できるので、傾斜楕円を2つの連立漸化式で生成することができる。

なお、点列数 $N$ は偶数でなければならず、 $m$ を任意の整数として、 $N = 8m$ 、 $N = 8m + 4$ 、 $N = 4m + 2$ の場合とで生成アルゴリズムが異なる。このため、図2に示すような場合分けが必要である。ここで、 $N = 8m$ の場合の楕円生成アルゴリズムを示す。

$N = 8m$ の場合

[ Initialize ]

$$\Delta\theta = 4\pi/N; \varepsilon = 2 \sin(\Delta\theta/2)$$

$$x_0 = a \cos \phi; y_0 = a \sin \phi$$

$$x'_0 = a \cos \phi \sin(\Delta\theta/2) + b \sin \phi \cos(\Delta\theta/2)$$

$$y'_0 = a \sin \phi \sin(\Delta\theta/2) - b \cos \phi \cos(\Delta\theta/2)$$

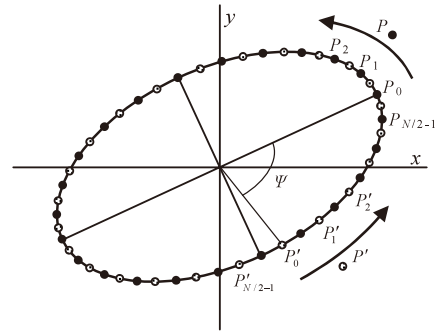
[ Loop ]  $n = 0, 1, 2, \dots, N/2 - 1$

$$\text{plot}(x_c + x_n, y_c + y_n); \text{plot}(x_c + x'_n, y_c + y'_n)$$

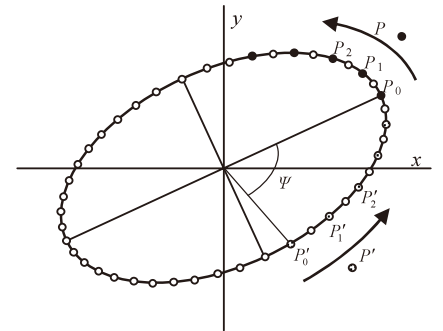
$$x_{n+1} = x_n - \varepsilon \cdot x'_n; x'_{n+1} = x'_n + \varepsilon \cdot x_{n+1}$$

$$y_{n+1} = y_n - \varepsilon \cdot y'_n; y'_{n+1} = y'_n + \varepsilon \cdot y_{n+1}.$$

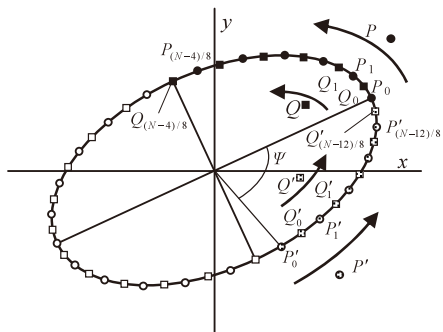
$N = 8m$  以外の場合も、上記アルゴリズムに若干



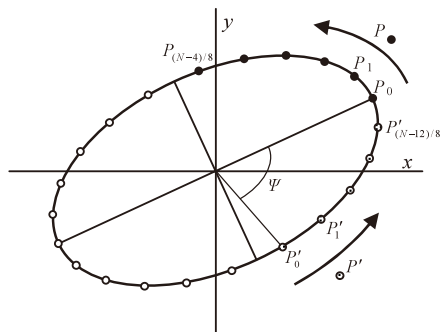
(a)  $N=8m$



(b)  $N=8m+4$



(c)  $N=8m+4$



(d)  $N=4m+2$

図2 輿水の楕円生成法

Fig. 2 Koshimizu's ellipse generating method.

の変更を加えるだけでよい。なお、 $N$  の場合分けによらず、1 点を生成するための本手法の乗算回数は 2 回である。

### 3. 三項漸化式を用いた傾斜楕円生成法

本章では、三角関数の間の三項漸化関係を利用した傾斜楕円生成法について述べる。

#### 3.1 三角関数の 2 つの三項漸化式に基づく手法

輿水の手法では、 $x_n$  と  $x'_n$  とでパラメータ  $\theta_n$  の位相が  $-\pi/2 + \Delta\theta/2$  もずれていたため、取扱いが難しかった。このような問題を解決するには、パラメータ  $\theta_n$  の位相が互いに大きくずれている  $x_n$  と  $x'_n$  とから  $x_{n+1}$  を計算するのではなく、位相が刻み角  $\Delta\theta$  だけ異なる  $x_n$  と  $x_{n+1}$  とから  $x_{n+2}$  を計算する方法に変更すればよい。このような 3 つの項からなる漸化式、すなわち三項漸化式を導くため、まず次の余弦関数の定義式を考える。

$$e^{i\Delta\theta} + e^{-i\Delta\theta} = 2 \cos \Delta\theta. \tag{8}$$

ここで、両辺に  $e^{i\theta_n}$  を乗じれば次式を得る。

$$e^{i\theta_{n+1}} + e^{-i\theta_{n-1}} = 2 \cos \Delta\theta \cdot e^{i\theta_n}. \tag{9}$$

上式の実数部と虚数部から、以下の 2 つの三項漸化式が得られる。

$$\left. \begin{aligned} \cos \theta_{n+1} + \cos \theta_{n-1} &= 2 \cos \Delta\theta \cdot \cos \theta_n \\ \sin \theta_{n+1} + \sin \theta_{n-1} &= 2 \cos \Delta\theta \cdot \sin \theta_n \end{aligned} \right\} \tag{10}$$

上式で示した三角関数の 2 つの三項漸化式は同形であるから、これを利用することにより、式 (4) から次式の三項漸化式を得る。ただし、 $\alpha = 2 \cos \Delta\theta$  とする。

$$\left. \begin{aligned} x_{n+2} &= \alpha \cdot x_{n+1} - x_n \\ y_{n+2} &= \alpha \cdot y_{n+1} - y_n \end{aligned} \right\} \tag{11}$$

$x_n$  と  $y_n$  は  $x_0, x_1$  および  $y_0, y_1$  が初期値として与えられれば、上式の三項漸化式で順次計算できる。また、初期値も式 (4) の傾斜楕円の定義式に従って計算するだけでよいから、Smith の手法の初期化条件と比べて簡単である。以下に、傾斜楕円生成アルゴリズムを示す。

[ Initialize ]

$$\begin{aligned} \Delta\theta &= 2\pi/N; \quad c = \cos \Delta\theta; \quad s = \sin \Delta\theta \\ \gamma &= \cos \phi; \quad \sigma = \sin \phi; \quad \alpha = 2c \\ x_0 &= a\gamma; \quad x_1 = cx_0 - bs\sigma; \quad y_0 = a\sigma; \quad y_1 = cy_0 + bs\gamma \\ &\text{plot}(x_c + x_0, y_c + y_0) \end{aligned}$$

[ Loop ]  $n = 0, 1, 2, \dots, N - 2$

$$\begin{aligned} &\text{plot}(x_c + x_{n+1}, y_c + y_{n+1}) \\ &x_{n+2} = \alpha \cdot x_{n+1} - x_n \end{aligned}$$

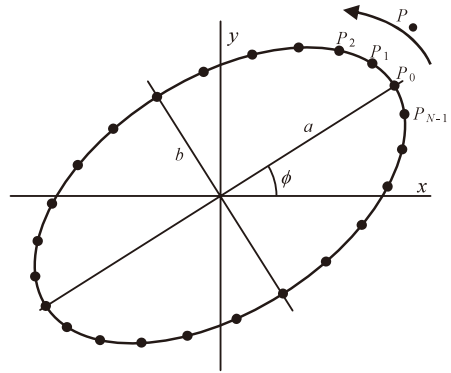


図 3 三項漸化式を用いた楕円の発生  
Fig. 3 Proposed ellipse generation by using three-term recurrence.

$$y_{n+2} = \alpha \cdot y_{n+1} - y_n.$$

本手法で発生した点列を図 3 に示す。これらの点列は、Smith の手法や輿水の手法で発生する傾斜楕円の点列と同一である。ただし、連立漸化式を用いた輿水の手法では点列数  $N$  が偶数でなければならず、さらに計算アルゴリズムも  $N$  により 3 つの場合分けが必要であった。これに対し、提案手法では  $N$  が奇数であってもよく、また 8 の倍数などの場合分けも必要でない。

#### 3.2 楕円弧の生成法

傾き  $\phi$  の傾斜楕円において、開始角  $\phi_1$ 、終了角  $\phi_2$  の楕円弧を  $N$  個のドットで生成させることを考える。輿水の手法では、連立漸化式で発生させた 2 点間の角度パラメータの位相が互いに  $\pi/2 - \Delta\theta/2$  だけずれている。このため、楕円の 1/4 以下の長さの楕円弧では、求める楕円弧以外の部分の計算も行わなければならない。これに対し、提案手法では点列を楕円の経路に従って順次発生できるので、始点として楕円弧の端点を選べば、効率良く楕円弧を発生できる。

さて、角度パラメータ  $\theta$  と実際の角度  $\theta'$  との間には、

$$\tan \theta' = \frac{b}{a} \tan \theta \tag{12}$$

の関係がある。これより、開始角  $\phi_1$ 、終了角  $\phi_2$  にそれぞれ対応する角度パラメータを  $\phi_1^*, \phi_2^*$  とすると、

$$\left. \begin{aligned} \phi_1^* &= \tan^{-1} \left( \frac{a}{b} \tan \phi_1 \right) \\ \phi_2^* &= \tan^{-1} \left( \frac{a}{b} \tan \phi_2 \right) \end{aligned} \right\} \tag{13}$$

となる。角度パラメータの増分を  $\Delta\theta = (\phi_2^* - \phi_1^*) / (N - 1)$ 、 $\alpha = \cos 2\Delta\theta$  とすることにより、提案手法を用いて楕円弧を生成することができる。これを図 4 に示す。

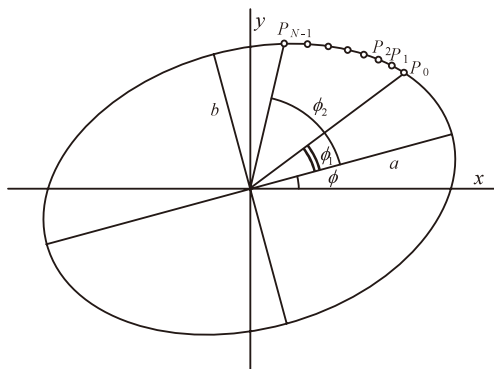


図 4 三項漸化式を用いた楕円弧の発生

Fig. 4 Arc-ellipse generation by using three-term recurrence.

表 1 楕円の 1 点あたりの計算回数

Table 1 Calculation counts of a point on ellipse.

	Smith	Koshimizu	Proposed method
Multiplication	4	2	2

表 2 楕円弧の 1 点あたりの計算回数

Table 2 Calculation counts of a point on arc-ellipse.

	Smith	Koshimizu	Proposed method
Multiplication	4	4	2

### 3.3 計算コスト

Smith の手法，輿水の手法，提案手法の各々の計算量を，乗算回数から求めることにしよう．表 1 に示すように，楕円の場合は Smith の手法に比べ，輿水の手法，提案手法ともに乗算回数は半分になっている．このため，計算コストはおおよそ 1/2 になる．

長さが楕円の 1/4 以下である楕円弧 1 点あたりの乗算回数は，表 2 に示すように，Smith の手法も輿水の手法も同じである．なぜなら，長さが楕円の 1/4 以下である楕円弧では，輿水の手法で発生させた点列のうち半分が求める楕円弧の外側の点であるためである．これに対し，提案手法の乗算回数は Smith の手法，輿水の手法の半分である．

## 4. 実 験

本章では，Smith の手法，輿水の手法，提案手法とで，傾斜楕円と楕円弧の生成における計算速度を比較する．

### 4.1 傾斜楕円と楕円弧の生成

傾斜楕円の主軸半径を  $a = 21$ ，副軸半径を  $b = 12$ ，傾斜角を  $\phi = 25$  度，構成点数を  $N = 100$  として傾斜楕円を生成する．ここで，傾斜楕円の点対称性を利

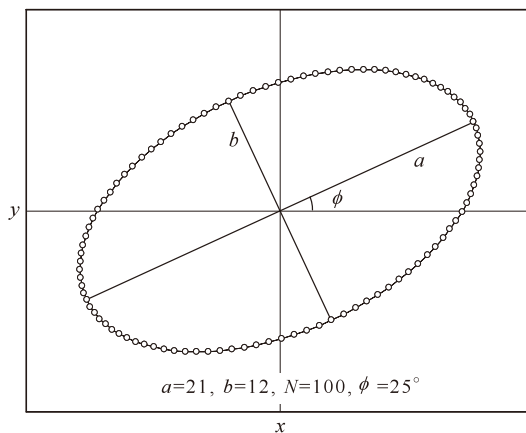


図 5 提案手法による楕円生成

Fig. 5 Generated ellipse by using proposed method.

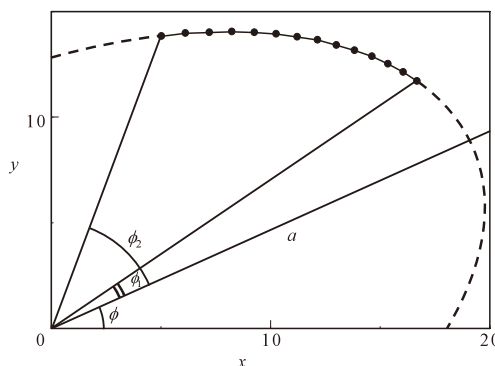


図 6 提案手法による楕円弧計算

Fig. 6 Generated arc-ellipse by using proposed method.

用し， $x, y$  とともに半分のみを計算し，残りの点は点対称に拡張して得た．また，計算は単精度実数で行った．図 5 は生成した楕円である．3 手法とも生成点は同一であった．

また，図 6 の楕円弧は，主軸半径を  $a = 21$ ，副軸半径を  $b = 12$ ，傾斜角を  $\phi = 25$  度，始点角度を  $\phi_1 = 10^\circ$ ，終点角度を  $\phi_2 = 45^\circ$  とし，構成点数は  $N = 14$  である．計算は単精度実数で行った．3 手法とも生成点は同一であった．

### 4.2 丸め誤差の影響

理論上は 3 手法とも同一点が得られるが，漸化式を用いているため，丸め誤差が累積すると考えられる．そこで，図 7 に，主軸半径  $a$  を 10 から 100 まで 10 ずつ変化させ，同時に傾斜角  $\phi$  をランダムに 1,000 回変化させた場合の計算誤差の最大値を調べた．副軸半径を  $b = 12$ ，生成点数を  $N = 100$  とした．計算は  $x, y$  とともに半分の 50 点だけ計算し，傾斜楕円の点対称性を利用して点対称に拡張した．また，変数を単精度実

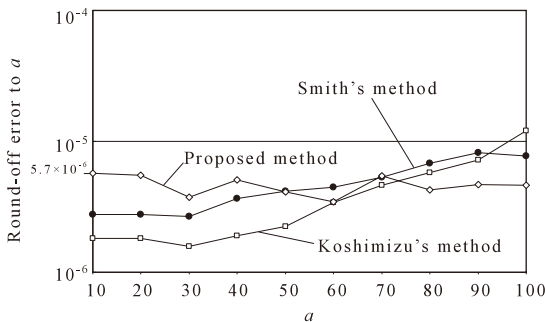


図7 楕円計算における丸め誤差  
Fig. 7 Round-off errors in ellipse generation.

表3 楕円の計算時間 (100点)

Table 3 Calculation time of ellipse (100 points).

Methods	Trigonometric	Smith	Koshimizu	Proposed method
Time(ns)	24585	1573	781	748

CPU: Pentium4 2.4GHz

数で扱った。  $a = 10$  のときは輿水の手法, Smith の手法, 提案手法の順に計算誤差が小さいが,  $a$  が増えるに従いその差は小さくなり,  $a = 70$  では3手法とも計算誤差がほぼ一致した。  $a = 100$  では逆に提案手法, Smith の手法, 輿水の手法の順に計算誤差が小さくなったが, 3手法とも大差はなかった。提案手法の主軸半径  $a$  に対する誤差の割合は, 最大で  $5.7 \times 10^{-6}$  であった。単精度演算の最小単位であるマシンイプシロンが  $2^{-23} \approx 1.19 \times 10^{-7}$  であることを考えると, 十分に実用的な精度と考えられる。

#### 4.3 計算時間

図5の傾斜楕円の計算に要する時間を, 3手法および式(2)の一般的な三角関数を用いた手法とで調べ, 表3に示した。傾斜楕円の点対称性より,  $x, y$  ともに半分の点だけを計算した。楕円1個あたりの計算時間は, 傾斜楕円を100万個連続で計算し, これを1ms単位で計測, その平均値として1ns単位で求めてある。また, CPUにPentium4-2.4GHzを搭載したパソコンを用い, プログラムはVC/C++, 変数は単精度実数で扱った。なお, 描画出力装置の種類により傾斜楕円の出力時間が異なるため, 計算時間には描画のための出力時間を含めていない。表3より, 三角関数を用いた手法に比べSmithの手法は約16倍速く, 輿水の手法, 提案手法のいずれもSmithの手法よりも約2倍速いことが分かる。このことから, 傾斜楕円生成における輿水の手法, 提案手法の計算量はSmithの手法の計算量の半分であることが確かめられた。なお, 輿水の手法と提案手法とでは大差なかった。

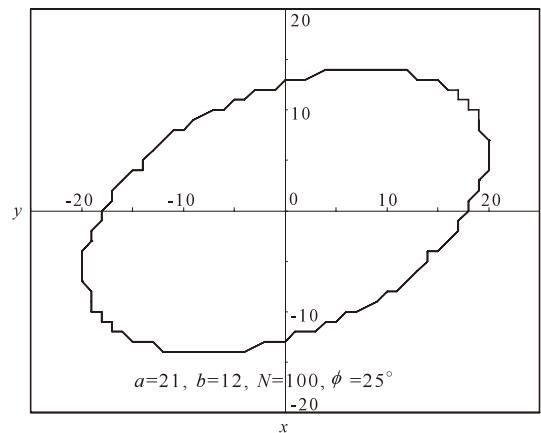


図8 ドット単位での楕円生成  
Fig. 8 Generated ellipse for dot matrix display.

表4 楕円弧の計算時間 (14点)

Table 4 Calculation time of arc-ellipse (14 points).

Methods	Trigonometric	Smith	Koshimizu	Proposed method
Time(ns)	6418	463	422	207

CPU: Pentium4 2.4GHz, 14points,  $\phi_1=10^\circ$ ,  $\phi_2=45^\circ$

なお, 出力装置をグラフィックスディスプレイとして出力時間を計測すると, グラフィックスメモリへの楕円1個の描きこみに要する出力時間は平均で1842nsであった。図8に, 図5の傾斜楕円をドット単位で描画した出力を示す。描画点の隣接点どうしは4連結で結ばれている。

表4は図6の楕円弧を, 3手法および三角関数を用いた手法で計算した時間である。描画のための出力時間は含まれていない。計算時間は連続で100万回楕円弧を発生させて, その平均値として求めた。三角関数を用いた手法に比べるとSmithの手法は約14倍速かったが, 輿水の手法はSmithの手法とは大差なかった。提案手法はSmithの手法と輿水の手法のいずれよりも約2倍速かった。このことから, 楕円弧生成における提案手法の計算量が, Smithの手法と輿水の手法の半分であることが確かめられた。

#### 5. おわりに

楕円をパラメトリック表現する立場から, 三角関数の2つの三項漸化式が同形であることを利用した, 数学的に厳密な傾斜楕円の高速計算法を提案した。

提案手法は初期設定や楕円弧描画が従来法と比較してシンプルであり, それを従来の高速生成法と同等以上の計算速度で実現した。また, この手法は楕円に内

接する多角形の面積を最大にするという意味において、最も描画品質に優れた Smith の手法や輿水の手法と同一の点列を生成することができる。しかも、数値計算にともなう丸め誤差は、Smith の手法、輿水の手法とほぼ同等である。

本手法は、楕円図形を扱うプロッタや図形処理、数値制御の高速化を可能にする。また、離散的フーリエ変換のように、円や楕円の各位相における  $x$  成分、 $y$  成分を含む数値計算に応用すると、直接三角関数を計算しなくてよいため高速処理が可能になる。特に、2 のべき乗でないデータ数の離散的フーリエ変換の計算には効果的である。

### 参 考 文 献

- 1) Pitteway, M.: Algorithm for drawing ellipses or hyperbolas with a digital plotter, *Computer J.*, Vol.10, No.3, pp.282-289 (1967).
- 2) 鴻田五郎, 清水賢資: 任意の楕円の高速ドット発生アルゴリズム, 電子通信学会論文誌 D, Vol.J68-D, No.11, pp.1934-1940 (1985).
- 3) 浅香孝雄, 山口雄二: 高速二次曲線図形発生器, 横河技報, Vol.27, No.3, pp.118-123 (1983).
- 4) Aken, V.: An Efficient Ellipse-Drawing Algorithm, *IEEE CG & A*, Vol.4, No.9, pp.24-35 (1984).
- 5) IBM Corp.: General method for drawing ellipses on raster graphics devices, *IBM Technical Disclosure Bulletin*, Vol.29, No.3, pp.1323-1327 (1986).
- 6) Fellner, D.W. and Helmborg, C.: Best Approximate General Ellipses on Integer Grids, *Comput. & Graphics*, Vol.18, No.2, pp.143-151 (1994).
- 7) 柿下尚武, 穂坂 衛: インクレメンタルな曲線の発生, 昭和 47 年度情報処理学会第 14 回大会予稿集, pp.267-268 (1972).
- 8) Smith, L.B.: Drawing ellipses, hyperbolas and parabolas with a fixed number of points and maximum inscribed area, *Comput. J.*, Vol.14, No.1, pp.81-86 (1971).
- 9) Prosser, C.J. and Alistair, C.K.: An Integer Method for The Graphical Output of Conic Sections, *ACM Trans. Graphics*, Vol.2, No.3, pp.182-191 (1983).
- 10) Salomon, D.: DDA Methods in TEX, *TUGboat*, Vol.10, No.2, pp.207-216 (1989).
- 11) Hearn, D. and Baker, J.P.: *Computer Graphics*, Prentice-Hall, Englewood Cliffs, NJ. (1986).
- 12) Rogers, D.F. and Adams, J.A.: *Mathematical Elements for Computer Graphics*, 2nd ed., McGraw-Hill, New York (1989).
- 13) 輿水大和, 沼田宗敏: 一般楕円生成の高速化について, 1993 年度電子情報通信学会春季大会, D-613, pp.73-74 (1993).
- 14) Neal, L.R. and Pitteway, M.: Yet More Circle Generators, *Computer J.*, Vol.33, No.5, pp.408-411 (1990).

(平成 19 年 1 月 16 日受付)

(平成 19 年 9 月 3 日採録)



沼田 宗敏 (正会員)

1984 年富山大学理学部物理学科卒業。2006 年富山県立大学大学院工学研究科博士後期課程修了 (社会人特別選抜)。博士 (工学)。1984 年 (株) ロゼフテクノロジーに入社し現在に至る。主たる研究テーマは、人工知能、機械学習、コンピュータグラフィックス、3 次元データ処理、画像処理。電子情報通信学会、精密工学会各会員。共著書に、『最新コンピュータグラフィックスがわかる』(技術評論社) 等。



輿水 大和 (正会員)

1975 年名古屋大学大学院工学研究科博士課程修了。工学博士。名古屋大学助手, 名古屋市立工業研究所を経て, 1986 年中京大学教養部教授, 1990 年同情報科学部教授, 2004 年同情報科学部長, 2006 年より情報理工学部長。画像処理, マシンビジョン, パターン認識, 顔研究, 画像デジタル化理論, Hough 変換等の画像処理基礎理論, およびそれらの産業応用研究に従事。電気学会 (上級会員, 協同研究委員会委員長), 電子情報通信学会 (教科書委員), SICE (PM 部会顧問), JSPE (IAIP 副委員長), 日本顔学会 (理事), 各種国内・国際学会 (SSII, ViEW, DIA/QCAV, FCV 等) で活動中。共著書に、『画像処理基本技法』(技術評論社), 『コンピュータビジョン』(近代科学社), 『実践画像処理』(Springer-Verlag 東京), 『信号処理』(オーム社) 等。ViEW2002 小田原賞 (IAIP, JSPE), ViEW2005 小田原賞 (IAIP, JSPE) 等受賞。



秦野 甯世 (正会員)

1972年北海道大学大学院理学研究科博士課程修了。理学博士。同年名古屋大学大型計算機センター助手。現在、中京大学情報理工学部教授。コンピュータシミュレーションと可視化、数学ソフトウェアの開発研究に従事。日本応用数理学会、日本化学会各会員。



野村 俊 (正会員)

1975年富山大学大学院工学研究科修士課程修了。工学博士(東京工業大学)。現在、富山県立大学工学部教授。日本光学会(応用物理学会)、精密工学会、日本機械学会、先端加工学会、Optical Society of America、American Society for Precision Engineering 各会員。共著書に、『インプロセス計測・制御・加工』(日刊工業新聞社)等。



神谷 和秀 (正会員)

1992年富山大学大学院工学研究科修士課程修了。博士(工学)(東京大学)。現在、富山県立大学工学部准教授。日本光学会(応用物理学会)、精密工学会、日本機械学会、先端加工学会、Optical Society of America、American Society for Precision Engineering 各会員。



二宮 市三 (正会員)

1921年生。1943年東京帝国大学工学部航空学科機体専修卒業。工学博士。1945年より40年間名古屋大学工学部に奉職、同大学大学院工学研究科情報工学専攻教授を経て1985年定年退職。1980年情報処理学会創立25周年記念論文賞受賞。主たる研究テーマは、ブール関数の構造、数値解析、数値計算ライブラリ。日本応用数理学会会員。