情報処理学会論文誌 Vol. 49 No. 9 3316–3326 (Sep. 2008)

正弦三項漸化式による円と楕円の高速生成法

沼田宗敏[†1] 輿水大和[†2] 秦野甯世[†2]
神谷和秀[†3] 野村俊[†3] 二宮市三[†4]

先に著者らは，余弦三項漸化式を用いたパラメトリック方式による効率的な傾斜楕円生成法を提案した．このような効率的な生成法は傾斜楕円よりもむしろ，使用頻度の高い真円や普通楕円でより強く求められている．本論文では，1 点あたり 2 回の乗算で真円および普通楕円を生成できる正弦三項漸化式を用いた手法を提案する．初期設定や弧の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，理論誤差がなく丸め誤差も余弦三項漸化式よりも小さい．

Fast Algorithm for Generating Circle and Ellipse by Using Sine-three-term-recurrence

Munetoshi Numada,[†1] Hiroyasu Koshimizu,[†2]
Yasuyo Hatano,[†2] Kazuhide Kamiya,[†3]
Takashi Nomura[†3] and Ichizo Ninomiya[†4]

We proposed an efficient algorithm for generating oblique ellipse by using the parametric equation called CTTR (cosine-three-term-recurrence). However, the efficient generating method for circle and normal ellipse is strongly required rather than the oblique ellipse. Thus this paper proposes a new sine-three-term recurrence algorithm for generating circles and normal ellipses with only two multiplications per point. Since the proposed method is simpler than the traditional methods both in setting initial conditions and in generating arcs, this method can be superior or equivalent to those of the traditional methods with respect also to the computation cost. Numerical experiments have shown its higher quality both in speed and accuracy. In addition it is notable that this method has no approximation error, and that the round-off error is smaller than CTTR.

1. はじめに

円や楕円等基本的な図形の計算手法には判別関数を用いるノンパラメトリック方式と，角度パラメータを用いるパラメトリック方式とがある．前者の方式は[1)–8)]，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，また生成ドットを整数で扱うのでグラフィックスディスプレイへの応用に適している．これに対し後者の方式では次のように処理を行う[9)–13)]．まず最初に図形の粗い点列を生成する．続いて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[14)]．図形を構成する点列を実数で扱うため，プロッタや数値制御に適している．

パラメトリック方式では，計算コストの大きな三角関数の計算を初期値計算だけに限定し，あとは漸化式を用いて点列を順次計算する．効率化の鍵はこの漸化式における乗算回数の低減にある．このような観点から，筆者らは一対の余弦三項漸化式 CTTR（cosine-three-term-recurrence）を用いた効率的な傾斜楕円の点列生成手法を提案した[15)]．この手法には理論誤差がなく，1 点を計算するために必要な乗算は 2 回と少ない．しかしながら，このような効率化はむしろ傾斜楕円よりも使用頻度の高い真円や傾きのない普通楕円において，より強く求められている．

パラメトリック方式における，真円および普通楕円の高速な点列生成手法は以下のとおりである．真円生成では Neal らの計算手法がある[9)]．この手法は 1 点を計算するために乗算を 3 回必要とする．また，計算値に理論誤差がともなわず，微小値を加減算する形の漸化式であるため丸め誤差が累積しにくい．普通楕円生成では Smith の計算手法がある[11)]．この手法は普通楕円だけでなく傾斜楕円も生成できる．理論誤差がなく，1 点を計算するために必要な乗算は 4 回である．

ところで，傾き角が 0 のとき傾斜楕円は普通楕円になり，さらに主軸半径と副軸半径とを等しくすれば真円となる．これより，CTTR を用いた傾斜楕円生成法は普通楕円や真円にも適

†1 株式会社ロゼフテクノロジー
Lossev Technology Corporation
†2 中京大学
Chukyo University
†3 富山県立大学
Toyama Prefectural University
†4 名古屋大学名誉教授
Professor Emeritus, Nagoya University

用できる．この場合，1 点あたり 2 回の乗算で計算できるので，従来法に比べて乗算回数を大幅に削減できる．しかしながら，この手法の漸化式は引き算を主用する形となっているので，丸め誤差が累積しやすい．このため，1 点あたりの乗算回数が少ないだけでなく，丸め誤差も累積しにくい計算手法への改良が求められている．Neal らの真円生成手法のような微小値を加減算する形式の漸化式を採用することにより，この問題を解決することができると思われる．

そこで本論文では，余弦三項漸化式 CTTR の代わりに正弦三項漸化式を導入し，これを用いた真円および楕円の点列生成手法を提案する．この手法は数学的に厳密な計算手法で，丸め誤差も従来法より小さい．しかも 1 点あたりの乗算は 2 回である．本論文の構成は以下のとおりである．2 章では真円および普通楕円の高速点列生成手法の関連研究について述べる．3 章では正弦三項漸化式を用いた真円および楕円の点列生成手法を提案する．4 章では新手法の効果について検証する．そして，5 章で全体をまとめる．

2. 関 連 研 究

本章では，従来の真円および普通楕円生成の関連研究について概説する．なお，本研究では円や楕円をx-y直交座標系で扱い，便宜上その中心を座標系原点におく．中心が原点からオフセットした図形に対しては描画点の座標にそのオフセット量を加えればよいので，この前提は一般性を失わない．

2.1 真円の生成手法

ここでは半径 r の真円を，角度パラメータ $\theta_n = n\theta$ を用いて次式のように表現する．ここで，変数 n: $n = 0, 1, \cdots, N-1$ は図形を構成する総点数 N の番号であり，変数 $\theta = 2\pi/N$ は微小角である．

$$\left.\begin{aligned} x_n &= r\cos\theta_n \\ y_n &= r\sin\theta_n. \end{aligned}\right\} \tag{1}$$

上式は，加法定理によって次式の漸化式へと置き換えできる．初期値 $x_0 = r$，$y_0 = 0$，$\cos\theta$ および $\sin\theta$ の計算だけで，他の点は漸化式から順次計算できるので三角関数演算が不要になる．このとき，1 点を計算するための乗算は 4 回である．

$$\left.\begin{aligned} x_{n+1} &= x_n\cos\theta - y_n\sin\theta \\ y_{n+1} &= y_n\cos\theta + x_n\sin\theta. \end{aligned}\right\} \tag{2}$$

式 (2) の真円計算式において，$\cos\theta$ を $\varepsilon = \sin\theta$ に関する 1 次のマクローリン近似多項式で近似すると次式が得られる．

$$\left.\begin{aligned} x_{n+1} &= x_n - \varepsilon y_n \\ y_{n+1} &= y_n + \varepsilon x_n. \end{aligned}\right\} \tag{3}$$

上式を順次計算して得られる図形はスパイラルとなり，n が大きくなるにつれ発散する．しかし，2 行目右辺第 2 項の x_n を x_{n+1} に置き換えると発散しない形の次式を得る．コンピュータグラフィックスの黎明期には，真円はこの漸化式で近似的に計算された[9]．ここで，$\varepsilon = 2\sin(\theta/2)$ である．

$$\left.\begin{aligned} x_{n+1} &= x_n - \varepsilon y_n \\ y_{n+1} &= y_n + \varepsilon x_{n+1}. \end{aligned}\right\} \tag{4}$$

上式は x_n，y_n ともに右辺第 2 項の微小値を加減算する形式を持ち，丸め誤差累積の抑制に強い．しかも，1 点生成のために必要な乗算は 2 回と少ない．しかし，この式で計算される図形は真円よりもわずかに膨らんだ，$\pi/4$ の傾きを持つ傾斜楕円にすぎない．

これに対し，Hong は式 (2) の $\cos\theta$ を $\varepsilon = \sin\theta$ に関する 2 次式のマクローリン近似多項式で近似した[10]．これを次式に示す．ここで，$\beta = 1 - \varepsilon^2/2$ である．

$$\left.\begin{aligned} x_{n+1} &= \beta x_n - \varepsilon y_n \\ y_{n+1} &= \beta y_n + \varepsilon x_n. \end{aligned}\right\} \tag{5}$$

理論誤差は式 (4) と比べて十分に小さくなるが，それでも厳密な真円ではない．また，1 点あたりの乗算は 4 回となり式 (4) の 2 回に比べて逆に多くなっている．ただし，整数 m を用いて $\varepsilon = 2^{-m}$ となるように ε をうまく選ぶことができれば，4 回の乗算を 4 回のシフト演算に置き換えできるので高速化が期待できる．ところが，このような ε を選ぶには生成点数 N に大きな制約が必要である．このため，一般的な高速化手法とはいえない．このように，理論誤差の低減と乗算回数の低減とは簡単に両立しない．

なお，Hong の手法は並列演算形式であるため，ハードウェアによっては高速化が期待できる．上式をプログラム化すると，x_{n+1} の計算，y_{n+1} の計算に続いてループ制御命令がくる．このため，x_{n+1}，y_{n+1} の計算に必要な x_n，y_n は直前のステップで計算されない．したがって，スーパスケーラ（superscalar）方式の CPU を用いると，x_{n+1} の計算と y_{n+1} の計算とを同時処理できる．

一方，Neal らは式 (4) を改良し理論誤差のない真円の計算式を導いた[9]．

$$\left.\begin{aligned} x'_{n+1} &= x_n - ty_n \\ y_{n+1} &= y_n + Tx'_{n+1} \\ x_{n+1} &= x'_{n+1} - ty_{n+1}. \end{aligned}\right\} \tag{6}$$

ここで，$t = \tan(\theta/2)$，$T = 2t/(1+t^2)$ である．真円上の 1 点を生成する乗算も 3 回と少ない．しかしながら，式 (6) の 1 行目の x'_n はプロットに直接関係しない一時的変数である．このような変数を省略できればさらにステップ数を少なくできるので，式 (6) のままでは効率化が不十分である．なお，2 行目，3 行目の計算は必ずその直前行の計算結果が終わらないと実行できないため，並列計算に向かない直列演算形式となっている．

このように，これまでの真円生成法では理論誤差をなくすことと乗算回数の低減とを両立させることができていない．このため，理論誤差をともなわずかつ 1 点あたりの乗算回数が 2 回以下となるような真円の計算手法が求められている．

2.2　普通楕円の生成手法

次に，普通楕円の効率的生成について述べる．主軸半径 a，副軸半径 b の普通楕円は次式で表現できる．

$$\left.\begin{aligned} x_n &= a\cos\theta_n \\ y_n &= b\sin\theta_n. \end{aligned}\right\} \tag{7}$$

Smith の傾斜楕円生成手法では，普通楕円の点列を下の漸化式で厳密に生成できる[11]．

$$\left.\begin{aligned} x_{n+1} &= Ax_n + By_n \\ y_{n+1} &= Cx_{n+1} + Dy_n. \end{aligned}\right\} \tag{8}$$

ここで，$A = \cos\theta$，$B = -(a/b)\sin\theta$，$C = (b/a)\tan\theta$，$D = 1/\cos\theta$ である．1 点を生成するための乗算は 4 回である．なお，式 (8) は 2 行目の計算が前の計算結果を受けて行われるため直列演算形式である．

これに対し，CTTR を用いた次式の傾斜楕円生成法が提案された[15]．

$$\left.\begin{aligned} x_{n+2} &= \alpha x_{n+1} - x_n \\ y_{n+2} &= \alpha y_{n+1} - y_n. \end{aligned}\right\} \tag{9}$$

ここで，$\alpha = 2\cos\theta$ である．1 点を生成するための乗算は 2 回と少ない．しかも，1 行目，2 行目の計算はその後にループ制御命令をともなうので，直前行の計算結果が終わらなくても計算できる並列演算形式である．生成される傾斜楕円には理論誤差がともなわない．しかし，この漸化式は引き算を主用する形になっているので，丸め誤差が累積しやすい．そこで，3 章ではこの余弦三項漸化式 CTTR の代わりに正弦三項漸化式を導入し，丸め誤差の小さな計算手法を構築する．

3.　正弦三項漸化式による真円および普通楕円の生成手法

3.1　三項漸化式を用いた円と楕円の効率的な生成手法

指数関数による正弦関数の定義式

$$e^{i\theta} - e^{-i\theta} = 2i\sin\theta \tag{10}$$

の両辺に $e^{in\theta}$ を乗じると，$\theta_n = n\theta$ であることから，

$$e^{i\theta_{n+1}} - e^{i\theta_{n-1}} = 2i\sin\theta \cdot e^{i\theta_n} \tag{11}$$

となる．この実数部と虚数部とから次の一対の漸化式を得る．

$$\left.\begin{aligned} \cos\theta_{n+1} - \cos\theta_{n-1} &= -2\sin\theta\sin\theta_n \\ \sin\theta_{n+1} - \sin\theta_{n-1} &= 2\sin\theta\cos\theta_n. \end{aligned}\right\} \tag{12}$$

式の 2 つの成分は互いに絡み合って分離できない連立形で，乗数係数に正弦関数を含む．これを正弦三項漸化式 STTR（sine-three-term-recurrence）という．

STTR の両辺に半径 r を乗じれば，真円の定義式からただちに真円の生成式 (13) を得る．ここで，$c = 2\sin\theta$ とする．

$$\left.\begin{aligned} x_{n+1} &= x_{n-1} - cy_n \\ y_{n+1} &= y_{n-1} + cx_n. \end{aligned}\right\} \tag{13}$$

初期値 (x_0, y_0) と (x_1, y_1) は，式 (1) でそれぞれ $\theta_0 = 0$，$\theta_1 = \theta$ とおいて得ることができる．**図 1** に，提案手法で生成した真円を示す．点 P_n の座標は (x_n, y_n) である．式 (13) は式 (3) や式 (4) ともよく似た形をしており，1 点あたりの計算量もほとんど同じである．しかも，両式とは異なり理論誤差がともなわない．また，理論誤差の生じない Neal らの手法と比較しても，1 点あたりの乗算が 2 回と小さい．なお，本手法は直前行の計算結果がなくても計算できる並列計算方式である．

次に，STTR の第 1 式の両辺に主軸半径 a を，第 2 式の両辺に副軸半径 b を乗じ，普通

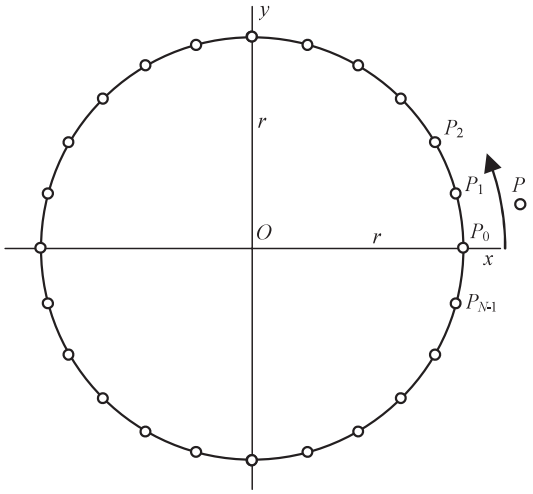


図 1　提案手法による真円生成
Fig. 1　Generated circle by using proposed method.

楕円の定義式を用いて $\cos\theta_n$，$\sin\theta_n$ を消去すれば普通楕円の生成式 (14) を得る．ここで，$c_1 = 2(a/b)\sin\theta$，$c_2 = 2(b/a)\sin\theta$ とする．

$$\left.\begin{aligned} x_{n+1} &= x_{n-1} - c_1 y_n \\ y_{n+1} &= y_{n-1} + c_2 x_n. \end{aligned}\right\} \tag{14}$$

初期値 (x_0, y_0) と (x_1, y_1) は，式 (7) でそれぞれ $\theta_0 = 0$，$\theta_1 = \theta$ とおいて得られる．**図 2** に，この漸化式で生成した普通楕円を示す．

式 (14) の漸化式の計算は分割数すなわち総点数 N があまり小さくなければ，式 (6) のように微小値の加減算を行うだけであるから桁落ちの恐れが小さい．

同様にして，指数関数による余弦関数の定義式

$$e^{i\theta} + e^{-i\theta} = 2\cos\theta \tag{15}$$

の両辺に $e^{in\theta}$ を乗じると，

$$e^{i\theta_{n+1}} + e^{-i\theta_{n-1}} = 2\cos\theta \cdot e^{i\theta_n} \tag{16}$$

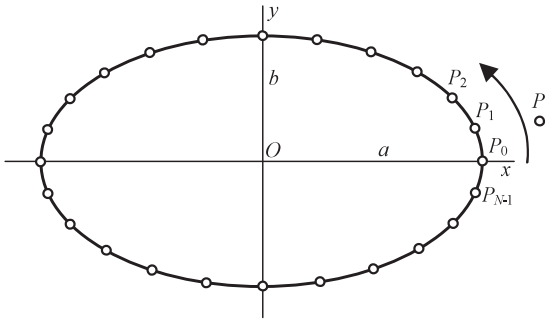


図 2　提案手法による普通楕円生成
Fig. 2　Generated normal ellipse by using proposed method.

となる．上式の実数部と虚数部から，次の一対の漸化式を得る．

$$\left.\begin{aligned} \cos\theta_{n+1} + \cos\theta_{n-1} &= 2\cos\theta\cos\theta_n \\ \sin\theta_{n+1} + \sin\theta_{n-1} &= 2\cos\theta\sin\theta_n. \end{aligned}\right\} \tag{17}$$

式の 2 つの成分は互いに独立かつ同形で，乗数係数に余弦関数を含む．これが文献 15) で導出した余弦三項漸化式 CTTR である．CTTR の両辺に半径 r を乗じれば，真円の定義式からただちに真円の生成式を得る．また，CTTR の第 1 式の両辺に主軸半径 a を，第 2 式の両辺に副軸半径 b を乗じれば，普通楕円の定義式からただちに普通楕円の生成式を得る．そして，これら 2 式に傾斜角 ϕ の回転を施せば，定義式 (18) から傾斜楕円の生成式 (9) を得る．

$$\left.\begin{aligned} x_n &= a\cos\phi\cos\theta_n - b\sin\phi\sin\theta_n \\ y_n &= a\sin\phi\cos\theta_n + b\cos\phi\sin\theta_n. \end{aligned}\right\} \tag{18}$$

式 (9) には，CTTR の 2 つの成分の同形性の効果が現れている．なお，式 (9) では分割数があまり小さくなければ現在の値の 2 倍程度の値から現在の値を引くという計算であるから，桁落ちの恐れが STTR を用いた式 (14) よりも大きくなる．

3.2　弧の生成法

弧の生成は，プロッタや数値制御において特に重要である．ここでは，構成点数を N，開始角を ϕ_1，終了角を ϕ_2 とし，また始点を (x_0, y_0)，終点を (x_{N-1}, y_{N-1}) として弧を生成することを考える．

まず，円弧の場合を考える．円弧は式 (13) の真円生成法において，初期値を (x_0, y_0)，

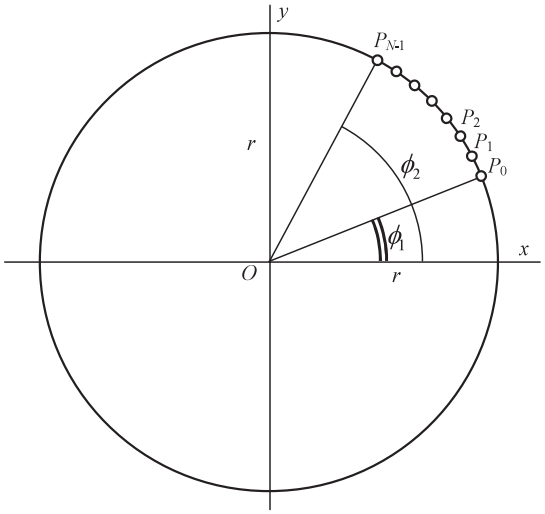


図 3　提案手法による円弧生成
Fig. 3　Generated circular arc by using proposed method.

(x_1, y_1) とおき，刻み角を $\theta = (\phi_2 - \phi_1)/(N-1)$ として与えられる．ここで，

$$\left.\begin{aligned} x_0 &= r\cos\phi_1, \quad x_1 = r\cos(\phi_1 + \theta) \\ y_0 &= r\sin\phi_1, \quad y_1 = r\sin(\phi_1 + \theta) \end{aligned}\right\} \tag{19}$$

である．**図 3** に提案手法で生成した円弧を示す．

続いて楕円弧について考える．角度パラメータ θ と実際の角度 θ' との間には，

$$\tan\theta' = \frac{b}{a}\tan\theta \tag{20}$$

の関係がある．これより，開始角 ϕ_1，終了角 ϕ_2 にそれぞれ対応する角度パラメータを ϕ_1^*，ϕ_2^* とすると，

$$\left.\begin{aligned} \phi_1^* &= \tan^{-1}\left(\frac{a}{b}\tan\phi_1\right) \\ \phi_2^* &= \tan^{-1}\left(\frac{a}{b}\tan\phi_2\right) \end{aligned}\right\} \tag{21}$$

となる．これより，初期値は次式で与えられる．

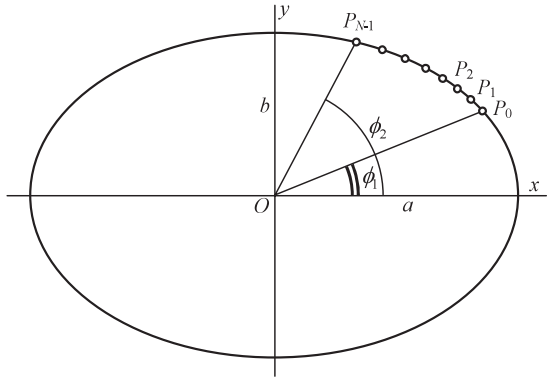


図 4　提案手法による楕円弧生成
Fig. 4　Generated elliptic arc by using proposed method.

$$\left.\begin{aligned} x_0 &= a\cos\phi_1^*, \quad x_1 = a\cos(\phi_1^* + \theta) \\ y_0 &= b\sin\phi_1^*, \quad y_1 = b\sin(\phi_1^* + \theta). \end{aligned}\right\} \tag{22}$$

角度パラメータの増分を $\theta = (\phi_2^* - \phi_1^*)/(N-1)$ とすると，普通楕円の弧を上式で計算できる．**図 4** に，提案手法で生成した楕円弧を示す．

3.3　真円および普通楕円の生成アルゴリズム

式 (13) による，中心 (x_0, y_0)，半径 r，構成点数 N の真円生成アルゴリズムを以下に示す．

［Initialize］

$\theta = 2\pi/N$;　$\sigma = \sin\theta$;　$c = 2\sigma$

$x_0 = r$;　$y_0 = 0$;　$x_1 = r\cos\theta$;　$y_1 = r\sigma$

$\mathrm{plot}(x_c + r, y_c)$

［loop］ $n = 1, 2, 3, \cdots, N-1$

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

$x_{n+1} = x_{n-1} - c \cdot y_n$

$y_{n+1} = y_{n-1} + c \cdot x_n$

式 (14) を用いた普通楕円生成アルゴリズムもほぼ同様である．

次に N を 8 の倍数にとったときの，真円の生成アルゴリズムを示す．まず，θ が $\pi/4$ の倍数になる点をあらかじめ与えておく．このとき，$0 < \theta < \pi/4$ の点を計算するだけで，$\pi/4 < \theta < \pi/2$ の点を x と y の入れ替えにより得ることができる．このため第 1 象限の点

の計算量は半分になる．さらに，x 軸と y 軸に対する線対称性を利用することにより，第 1 象限分の点の計算だけで残りすべての象限の点を線対称に拡張して得ることができる．よって，計算量を全体で 1/8 に減ずることができる．

［Initialize］

$\theta = 2\pi/N;\ \ \sigma = \sin\theta;\ \ c = 2\sigma;\ \ s = \sqrt{2}\,r/2$

$x_0 = r;\ \ y_0 = 0;\ \ x_1 = r\cos\theta;\ \ y_1 = r\sigma$

$\mathrm{plot}(x_c + r, y_c)$

$\mathrm{plot}(x_c + s, y_c + s)$

$\mathrm{plot}(x_c, y_c + r)$

$\mathrm{plot}(x_c - s, y_c + s)$

$\mathrm{plot}(x_c - r, y_c)$

$\mathrm{plot}(x_c - s, y_c - s)$

$\mathrm{plot}(x_c, y_c - r)$

$\mathrm{plot}(x_c + s, y_c - s)$

［loop］$n = 1, 2, 3, \cdots, N/8 - 1$

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

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

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

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

$\mathrm{plot}(x_c - x_n, y_c - y_n)$

$\mathrm{plot}(x_c - y_n, y_c - x_n)$

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

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

$x_{n+1} = x_{n-1} - c \cdot y_n$

$y_{n+1} = y_{n-1} + c \cdot x_n$

続いて，総点数 N を 4 の倍数にとったときの式 (14) による普通楕円生成アルゴリズムを示す．中心を (x_0, y_0)，主軸半径を a，副軸半径を b，構成点数を N とする．x 軸と y 軸に対する線対称性の利用により，第 1 象限分の点の計算だけで残りすべての象限の点を線対称に拡張して得ることができる．このため計算量は 1/4 になる．

［Initialize］

$\theta = 2\pi/N;\ \ \sigma = \sin\theta;\ \ c_1 = 2a\sigma/b;\ \ c_2 = 2b\sigma/a$

$x_0 = a;\ \ y_0 = 0;\ \ x_1 = a\cos\theta;\ \ y_1 = b\sigma$

$\mathrm{plot}(x_c + a, y_c)$

$\mathrm{plot}(x_c - a, y_c)$

$\mathrm{plot}(x_c, y_c + b)$

$\mathrm{plot}(x_c, y_c - b)$

［loop］$n = 1, 2, 3, \cdots, N/4 - 1$

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

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

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

$\mathrm{plot}(x_c - x_n, y_c - y_n)$

$x_{n+1} = x_{n-1} - c_1 \cdot y_n$

$y_{n+1} = y_{n-1} + c_2 \cdot x_n$

3.4　計算コスト

効率化の目安である真円および円弧の 1 点あたりの乗算回数を各手法別に調べた．これを**表 1** に示す．真円と円弧のいずれの場合でも，Hong の手法で 4 回，Neal らの手法で 3 回である．これに対し提案手法は 2 回と少ないので，Hong の手法に比べて処理時間が 0.5 倍，Neal らの手法に比べても 0.67 倍になると期待できる．

同様に，普通楕円および楕円弧の 1 点あたりの乗算回数を調べ，**表 2** に示した．普通楕円と楕円弧のいずれにおいても，Smith の手法で 4 回，CTTR を用いた手法では 2 回であっ

表 1　円・円弧の 1 点あたりの乗算回数
Table 1　Calculation counts of a point on circle and circular arc.

	Hong	Neal	Proposed method
Circle	4	3	2
Circular arc	4	3	2

表 2　楕円・楕円弧の 1 点あたりの乗算回数
Table 2　Calculation counts of a point on ellipse and elliptic arc.

	Smith	CTTR	Proposed method
Normal ellipse	4	2	2
Elliptic arc	4	2	2

た．STTR を用いた提案手法でも 2 回と少なく，Smith の手法に比べて 0.5 倍，CTTR を用いた手法とは同等の処理時間が期待できる．

4. 実　　験

本章では，真円・普通楕円および各々の弧の生成における計算誤差および計算速度を手法別に比較する．真円では Hong の手法，Neal らの手法，および STTR による提案手法を，普通楕円では Smith の手法，CTTR による手法および提案手法を用いた．

4.1 円・楕円および各々の弧の生成

半径を $r = 21$，主軸半径を $a = 21$，副軸半径を $b = 10$，構成点数を $N = 100$ として，提案手法により真円および普通楕円を生成する．構成点数は 4 の倍数であるので，x 軸と y 軸に対する線対称性を利用し，各々x，y ともに 1/4 の点だけを計算し，残りの点をそれぞれ線対称に拡張して得た．また，計算はすべて単精度実数で行った．**図 5** は提案手法で生成された真円である．Hong の手法，Neal らの 2 手法でも，提案手法の生成点とほぼ同一点が得られた．また，**図 6** は提案手法で生成した普通楕円である．Smith の手法，CTTR を用いた 2 手法でも，提案手法の生成点とほぼ同一点が得られた．

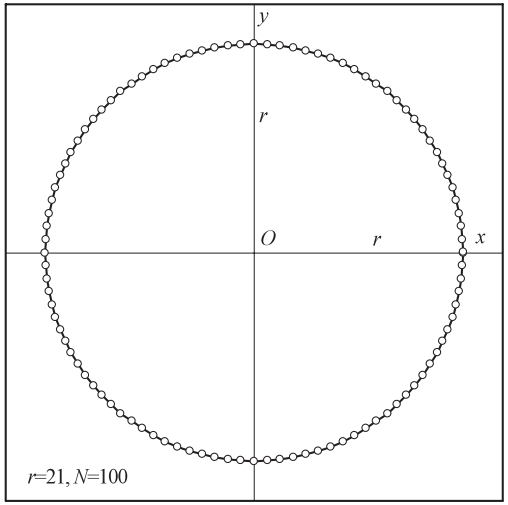


図 5　提案手法による真円生成
Fig. 5　Generated circle by using proposed method.

続いて，提案手法により円弧および楕円弧を生成する．半径を $r = 21$，主軸半径を $a = 21$，副軸半径を $b = 10$，始点角度を $\phi_1 = \pi/18$，終点角度を $\phi_2 = \pi/4$ とし，構成点数を $N = 14$ とした．**図 7** は提案手法で生成された円弧である．Hong の手法，Neal らの 2 手法でも，提案手法の生成点とほぼ同一点が得られた．**図 8** は提案手法で生成した楕円弧である．Smith

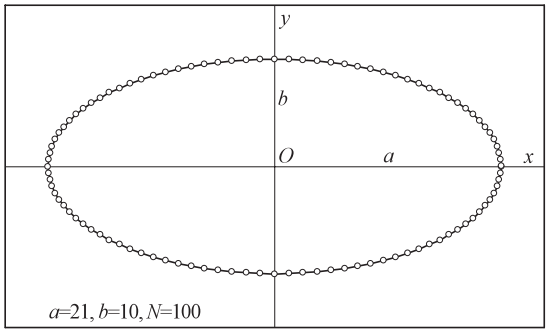


図 6　提案手法による楕円生成
Fig. 6　Generated ellipse by using proposed method.

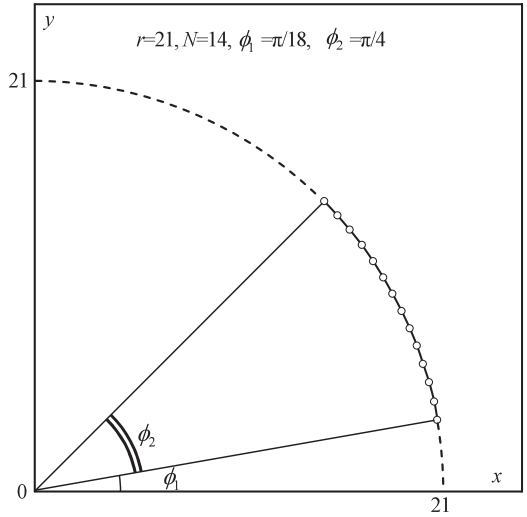


図 7　提案手法による円弧計算
Fig. 7　Generated circular arc by using proposed method.

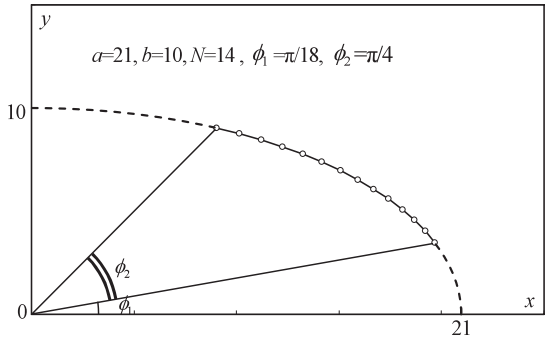


図 8　提案手法による楕円弧計算
Fig. 8　Generated elliptic arc by using proposed method.

の手法，CTTR を用いた 2 手法でも，提案手法の生成点とほぼ同一点を得ることができた．

4.2　計算誤差の影響

ここでは，真円および普通楕円における手法別の計算誤差について調べる．まず，真円生成の場合について調べる．Hong の手法では理論誤差が計算誤差として発生するが，Neal らの手法，提案手法はともに理論上真円に一致するため理論誤差が生じない．ただし，後者の場合には漸化式による繰返し演算で発生する丸め誤差が，計算誤差として現れる．そこで，真円の半径 r を 10 から 100 まで 10 ずつ変化させた場合の計算誤差の最大値を，r に対する比率で**図 9** に示した．生成点数は $N = 100$ である．真円の生成は線対称性を利用して，点総数の 1/4 だけを計算した．また，計算は単精度実数で行った．Hong の手法では，理論誤差は半径 r に対し 4.4×10^{-5} とほぼ一定であった．一方，理論誤差の発生しない Neal らの手法や提案手法ではこれより計算誤差が 2 桁も小さく，いずれも半径 r に対し 5.5×10^{-7} 以下であった．しかも，2 手法とも大差はなかった．単精度実数演算の最小単位であるマシンイプシロンが $2^{-23} \fallingdotseq 1.19 \times 10^{-7}$ 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．

次に，普通楕円生成における計算誤差について調べる．Smith の手法，CTTR を用いた手法，提案手法の 3 手法は，理論上いずれも普通楕円に一致するため理論誤差は発生しない．この場合には，漸化式の繰返し演算で発生する丸め誤差が計算誤差として現れる．ここで，**図 10** に主軸半径 a を 10 から 100 まで 10 ずつ変化させた場合の計算誤差の最大値を，a に対する比率で表した．副軸半径を $b = 10$，生成点数を $N = 100$ とした．真円の場合と

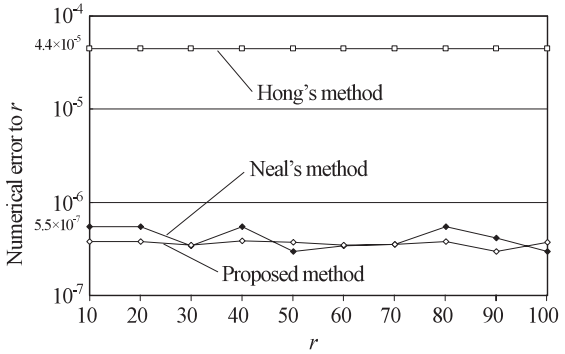


図 9　真円計算における計算誤差
Fig. 9　Numerical errors in circle generation.

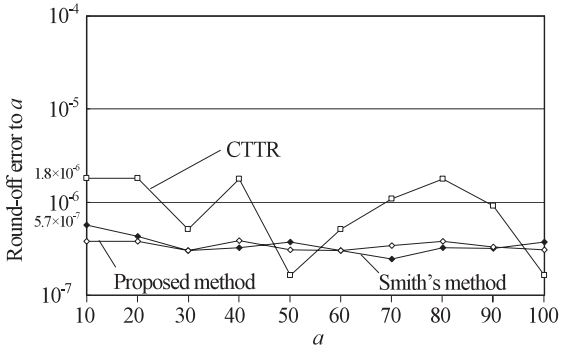


図 10　楕円計算における丸め誤差
Fig. 10　Round-off errors in ellipse generation.

同様，線対称性を利用し，点総数の 1/4 だけを単精度実数で計算した．CTTR による手法では，丸め誤差は半径 r に対し 1.8×10^{-6} 以下であった．また，Smith の手法と提案手法とは大差なく，いずれも 5.7×10^{-7} 以下であった．これは真円の 2 手法と同様，十分に実用的な精度である．なお，CTTR を用いた手法と提案手法が同じ三項漸化式であるにもかかわらず最大誤差に約 3 倍の開きがでた．このことから，前者では引き算を主用するため丸め誤差が大きく，後者では微小値を加減算する形式であるため丸め誤差が小さくなることが確かめられた．

表 3　真円の計算時間
Table 3　Calculation time of circle.

Methods	Hong	Neal	Proposed method
Circle	588	562	331
Arc	323	302	175

Time: (ns), Circle :100points, Arc: 14points, CPU: Pentium4 2.4GHz

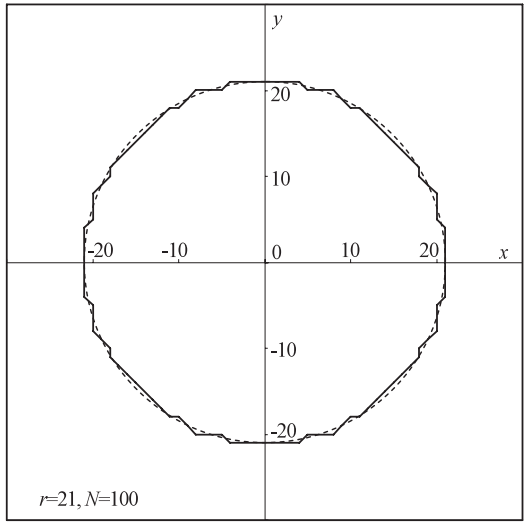
表 4　楕円の計算時間
Table 4　Calculation time of ellipse.

Methods	Smith	CTTR	Proposed method
Ellipse	783	354	340
Arc	435	190	179

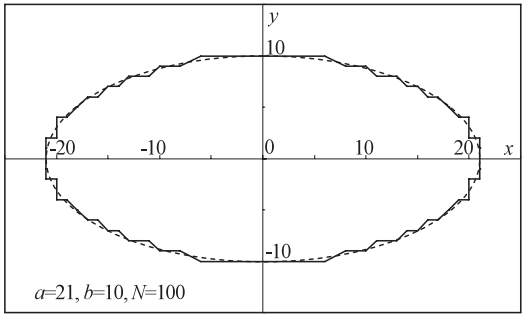
Time: (ns), Ellipse:100points, Arc: 14points, CPU: Pentium4 2.4GHz

4.3 計 算 時 間

図 5，図 7 の真円および円弧計算に要する時間を 3 手法で調べ，**表 3** に示した．真円では線対称性を利用して，x，y ともに点総数 $N = 100$ 個のうち 1/4 だけを計算した．これに対し円弧では全点数 $N = 14$ 個を計算した．1 個あたりの計算時間は 100 万個連続で計算しこれを 1 ms 単位で計測，その平均値として 1 ns 単位で求めてある．また，CPU にスーパスケーラ方式の Pentium4-2.4 GHz を搭載したパソコンを用い，プログラムは VC/C++，変数を単精度実数で扱った．なお，描画出力装置の種類により図形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．表より，提案手法の計算時間は Hong の手法の真円生成，円弧生成に比べて，それぞれ 0.56 倍，0.54 倍と短いことが分かる．これは，表 1 に示した両手法間の乗算回数の比率が，いずれも 0.5 倍であることを裏付けている．また，提案手法の計算時間は，Neal らの手法に比べて 0.59 倍，0.58 倍と短かった．これは，表 1 に示した乗算回数の比率がいずれも 0.67 倍であったのに比べて，比率が広がっている．Neal らの手法が並列計算向きではない直列演算形式であるのに対し，提案手法は並列演算形式であるためより高速化が実現できるためである．このため，2 手法の計算時間の比率が広がったものとみることができる．



(a)　Circle



(b)　Normal ellipse

図 11　ドット単位での円・楕円生成
Fig. 11　Generated circle and ellipse for dot matrix display.

次に，図 6，図 8 の普通楕円の計算に要する時間を 3 手法で調べ，**表 4** に示した．真円同様，普通楕円は点総数 $N = 100$ 点のうち 1/4 だけを計算し，楕円弧はすべての点 $N = 14$

点で計算した．楕円 1 個あたりの計算時間の測定条件は，真円の場合と同じである．表より，提案手法の計算時間は Smith の手法の普通楕円生成，楕円弧生成に比べて，それぞれ 0.43 倍，0.41 倍と短い．これは，表 1 に示した乗算回数の比率がいずれも 0.5 倍であったのに比べると，この比率が広がっている．提案手法が並列演算向きであるのに対し，Smith の手法が並列演算向きではないためである．また，提案手法の計算時間を CTTR による普通楕円生成，楕円弧生成と比べると，それぞれ 0.96 倍，0.94 倍とほぼ同じであった．両手法とも並列計算向きと同じ条件であるため，表 1 に示した両手法間の乗算回数の比率 1.0 倍に近い数値が得られている．

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円および楕円 1 個の描き込みに要する出力時間は平均で 1.8 μs であった．

図 11 に，図 5，図 6 の真円および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．

5. おわりに

パラメトリック表現する立場から，指数関数による正弦関数の定義式から導かれる一対の漸化式を用いた，数学的に厳密な真円および普通楕円の高速計算法を提案した．1 点あたりの計算における乗算は 2 回である．円弧や楕円弧の生成にも対応できる．提案手法は理論誤差がともなわないだけでなく，計算上の丸め誤差も従来法と同等かそれ以下である．そして，それを従来のどの高速生成法よりも高速に実現できた．

本手法の正弦三項漸化式 STTR は指数関数による正弦関数の定義式から導かれたが，式 (12) で $\theta = it$ とおくことにより，双曲線関数に対する一対の三項漸化式を得ることができる．この式を用いると，従来法よりも高速に双曲線の計算を行うことができる．

参 考 文 献

1) Horn, B.K.P.: Circle Generators for Display Devices, *CGIP5*, No.2, pp.280–288 (1976).
2) Suenaga, Y., Kamae, T. and Kobayashi, T.: A high-speed algorithm for the generation of straight lines and circular arcs, *IEEE Trans. C.*, Vol.C-28, No.10, pp.728–736 (1979).
3) Kulpa, Z.: A note on the paper by Horn, B.K.P.: Circle generators for display devices, *CGIP9*, pp.102–103 (1979).
4) Pitteway, M.: Algorithm for drawing ellipses or hyperbolas with a digital plotter, *Computer J.*, Vol.10, No.3, pp.282–289 (1967).
5) 釜江尚彦，小杉　信，星野肇夫：図形のドット表示，信学論，Vol.56-A，No.7, pp.401–408 (1973).
6) Aken, V.: An Efficient Ellipse-Drawing Algorithm, *IEEE CG & A*, Vol.4, No.9, pp.24–35 (1984).
7) IBM Corp.: General method for drawing ellipses on raster graphics devices, *IBM Technical Disclosure Bulletin*, Vol.29, No.3, pp.1323–1327 (1986).
8) Fellner, D.W. and Helmberg, C.: Best Approximate General Ellipses on Integer Grids, *Comput. & Graphics*, Vol.18, No.2, pp.143–151 (1994).
9) Neal, L.R. and Pitteway, M.: Yet More Circle Generators, *Computer J.*, Vol.33, No.5, pp.408–411 (1990).
10) Hong, T.: A High Precision Digital Differential Analyzer for Circle Generation, *NATO ASI Series*, Vol.F17, pp.239–256 (1985).
11) 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).
12) Hearn, D. and Baker, J.P.: *Computer Graphics*, Prentice-Hall, Englewood Cliffs, NJ. (1986).
13) Rogers, D.F. and Adams, J.A.: *Mathematical Elements for Computer Graphics, 2nd ed.*, McGraw-Hill, New York (1989).
14) 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).
15) 沼田宗敏，輿水大和，秦野甯世，神谷和秀，野村　俊，二宮市三：三角関数の三項漸化式による傾斜楕円の高速生成法，情報処理学会論文誌，Vol.48, No.12, pp.4051–4058 (2007).

(平成 20 年 2 月 7 日受付)
(平成 20 年 6 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．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，知能機械学，コンピュータグラフィックス，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，名古屋市工研を経て，現在中京大学情報理工学部長．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）等受賞．

秦野　甯世（正会員）

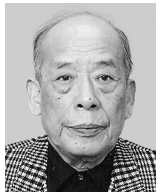
1972 年北海道大学大学院理学研究科博士課程修了．理学博士．同年名古屋大学大型計算機センター助手．現在，中京大学情報理工学部教授．コンピュータシミュレーションと可視化，数学ソフトウェアの開発研究に従事．日本応用数理学会，日本化学会各会員．

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．応用物理学会・日本光学会，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．2006 年文部科学大臣表彰・科学技術賞受賞．

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．応用物理学会・日本光学会，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．



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．近著に，『数値計算のつぼ』，『数値計算のわざ』（ともに共立出版）等．