同次化 NURBS 曲線*

服部岳士** 岩田茂人*** 藤原竹雄† 梶間博行*** 山口富士夫††

A Proposal of Homogenized NURBS Curves

Takeshi HATTORI, Shigeto IWATA, Takeo FUJIHARA, Hiroyuki KAJIMA, and Fujio YAMAGUCHI

The NURBS curves and surfaces have become the standard descriptions in the field of CAD and computer graphics. They have several problems, however, such as numerical instability, limited convex hull property, and inefficiency of computing points on a curve or a surface. In this paper, it is shown that these problems can be resolved by using our newly proposed *homogenized* NURBS. In the homogenized NURBS, the sum of the blending functions need not be unity and the fractional representation of the functions is replaced by the integral representation. By taking advantage of this characteristic and homogeneous curves, an extended finite difference method which efficiently and accurately computes points on a curve is presented. By using the homogenized NURBS, numerical instability vanishes, convex hull property holds strictly regardless of the signs of weights, and exact points on the curve can be computed very efficiently.

Key word : CAD/CAM, CAGD, rational curves and surfaces, homogeneous coordinates, homogeneous curves and surfaces, NURBS, finite difference method

1. はじめに

現在、図形・形状処理工学の応用分野である CAD において、自動車のボディや家電製品の外形等を表現するべく有理曲線・曲面が期待されている。NURBS は、有理曲線・曲面の代表的な表現形式である。NURBS は、有理 Bézier など他の表現形式に比べて形状表現能力が優れているため、CAD システムにおいて最も有望視されている。しかし NURBS にもいくつかの問題点がある。従来の NURBS では表現式が有理多項式であるため、その分母が 0 または、0 に近い点において数値的に不安定になりアルゴリズムが破綻することがある⁶⁾。また、有理多項式表現であるために除算による数値誤差が生じてしまう。

そこで本研究では、従来のユークリッド座標よりも次元高い同次座標⁵⁾で定義される同次曲線・曲面を対象にし、従来の NURBS に代わる、同次化 NURBS を提案する。そして、同次化 NURBS を用いることにより、完全整数演算による無誤差の点列算出を可能にした。また、算出速度の面においても差分演算を利用することにより、従来の方法よりも速く演算することが可能となった。

また、曲面は曲線を拡張したものと考えられるので、本文で論じた内容の多くが曲面にも当てはまる。

2. 同次曲線

2.1 同次曲線の定義

同次座標の定義より、2つの θ でない 3 次元ベクトルに対し、

$$\begin{bmatrix} X & Y & w \end{bmatrix} = k \begin{bmatrix} X' & Y' & w' \end{bmatrix} \quad (k \neq 0)$$

* 原稿受付 平成 9 年 12 月 12 日

** 学生会員 早稲田大学大学院 (東京都新宿区大久保 3-4-1)

*** 早稲田大学大学院

† シュルンベルジェ (株) (相模原市淵野辺 2-2-1)

†† 正 会 員 早稲田大学理工学部

この 2 つのベクトルは同値関係にあるとし、オーバーラインの記号を用いて $\begin{bmatrix} \overline{X} & \overline{Y} & \overline{w} \end{bmatrix} = \begin{bmatrix} \overline{X'} & \overline{Y'} & \overline{w'} \end{bmatrix}$ と表記する。この同値関係によって決まる同値類を点と呼ぶ。従って、平面同次曲線とは、2 次元射影空間 P^2 に付随するベクトル空間 (これを同次ベクトル空間と呼ぶ) のベクトル $P(t) = \begin{bmatrix} \overline{X(t)} & \overline{Y(t)} & \overline{w(t)} \end{bmatrix} \neq 0$ に同値関係を導入して定まる点の集合 $\overline{P(t)}$ であると定義される。

$$\overline{P(t)} = \begin{bmatrix} \overline{X(t)} & \overline{Y(t)} & \overline{w(t)} \end{bmatrix} \quad (1)$$

また、 $w(t) \neq 0$ を満たすすべての t に対して、次式により 2 次元ユークリッド空間のベクトル $p(t)$ が決まる。

$$p(t) = \begin{bmatrix} X(t)/w(t) & Y(t)/w(t) \end{bmatrix} \quad (2)$$

この $p(t)$ により決まる曲線は、有理曲線にほかならない。よって、 $w(t) \neq 0$ を満たすすべての t に対して、同次曲線と有理曲線との間に、上への一对一の写像が存在する。一方、 $w(t) = 0$ を満たす t が定める点は無限遠点であり、この点は従来の有理曲線では表現できない点である。また、fig. 1 から分かるように、同じ $p(t)$ を同次曲線で表現する場合、無数の表現式が存在する。

2.2 同次曲線の性質

同次曲線の主な性質を有理曲線と対応させて述べる。

(1) 表現式の自由度

有理曲線はアフィン不変性を保つため混ぜ合わせ関数 $N_{j,M}(t)$ の重心結合性 ($\sum N_{j,M}(t) = 1$) を必要としたが、同次曲線はこれを必要としない。なぜなら、同次座標ベクトルに対するアフィン変換 (より一般的には射影変換も含めて) は線形変換になるからである。したがってアフィン不変性のための重心結合性は不必要となる。

Fig. 1 Rational curve and homogenized curve

(2) 拡張された凸包性⁷⁾

重み係数に正負の値が混在する場合、有理曲線は凸包性を失うが、同次曲線は同次ベクトル空間において原点を頂点とするすい面として扱われるため（図 1 参照）、同次ベクトル空間において制御ベクトル方向の直線が凸包を張り、凸包性が成立する。

(3) 無限遠点

有理曲線 $p(t) = [X(t)/w(t) \ Y(t)/w(t)]$ は分母が 0 になる無限遠点で曲線は分断されてしまう。この無限遠点は、ユークリッド空間で取り扱うことができないうえに 0 で除算を行うため、例外処理が必要となる。これに対し、同次曲線は原点を頂点とする錐面を扱うため、曲線が分断されることはない。ユークリッド空間における無限遠点は、同次ベクトル空間では w 成分を 0 にするだけで他の有限点と同様に処理をすることができる。このため、処理において場合分けの必要がなくなり、単純なアルゴリズムを構築することができる。

(4) 非有理式表現による誤差のない数値演算と処理の安定性

有理曲線は有理多項式表現であるため、除算による数値誤差が生じてしまう。また、分母が 0 または 0 に近い点において数値的に不安定になる場合がある。しかし、同次曲線は非有理多項式表現であるため除算は行われない。曲線を同次ベクトル空間で扱うことにより係数を整数化でき、さらに除算による誤差を除くことができるので、厳密に正確な曲線上の点の算出が可能となる。また、有理曲線の場合、プログラムのいたるところに 0 による除算の危険が存在するが、同次曲線の場合、その危険がないため処理の安定性が向上する。

3. 同次化 NURBS 曲線

本章ではまず、従来の NURBS 曲線の定義を簡単に説明する。その後で、本研究で提案する同次化 NURBS を導出し、その幾何的性質について述べる。

3.1 通常の NURBS 曲線

2 次元 NURBS 曲線は、N 形式 B-Spline 関数⁴⁾ $N_{j,M}(t)$ と制御ベクトル q_j の線形結合で表現される。

$$p(t) = \frac{\sum_{j=0}^{n-1} N_{j,M}(t) w_j q_j}{\sum_{j=0}^{n-1} N_{j,M}(t) w_j} \quad (3)$$

ここに、

$$\cdot p(t) = [X(t)/w(t) \ Y(t)/w(t)] \in R^2$$

$$\cdot N_{j,M}(t) = \frac{t-t_j}{t_{j+M-1}-t_j} N_{j,M-1}(t) + \frac{t_{j+M}-t}{t_{j+M}-t_{j+1}} N_{j+1,M-1}(t)$$

$$N_{j,1}(t) = \begin{cases} 1 & (t_j \leq t < t_{j+1}) \\ 0 & (\text{上記以外}) \end{cases}$$

$$\cdot q_j = [x_j \ y_j] \in R^2 : \text{制御ベクトル}$$

$$\cdot w_j : \text{制御ベクトルの重み係数}$$

$$\cdot T = [t_0 \ t_1 \ t_2 \ \dots \ t_{M+n}] : \text{ノットベクトル}$$

$$\cdot M : \text{階数 (= 次数 + 1)}$$

$$\cdot n : \text{制御点数}$$

上式の $N_{j,M}(t)$ は de Boor Cox の漸化式とも呼ばれる。

第 1 章でも述べたように、通常の NURBS 曲線は有理式表現であるため、分母が 0 または 0 に近い点において数値的に不安定になってしまいアルゴリズムが破綻する等の問題がある。そこで、これらの問題点を解決するため次に述べる同次化 NURBS を提案する。

3.2 同次化 NURBS 曲線

ユークリッド空間で定義された NURBS (有理曲線) の問題を解決するため、NURBS を同次ベクトル空間で定義することを考える。さらに 2.2 節の同次曲線の性質 (1) を利用することにより、従来の NURBS に取って代わる同次化 NURBS を提案する。

制御ベクトルを同次ベクトル空間で定義し、混ぜ合わせ関数に N 形式 B-Spline 関数を用いたものを N 形式同次化 NURBS 曲線と呼び以下のように定義する。

$$P(t) = \sum_{j=0}^{n-1} N_{j,M}(t) Q_j \quad (4)$$

ここに、

$$\cdot P(t) = [X(t) \ Y(t) \ w(t)] \in R^3$$

$$\cdot N_{j,M}(t) = \frac{t-t_j}{t_{j+M-1}-t_j} N_{j,M-1}(t) + \frac{t_{j+M}-t}{t_{j+M}-t_{j+1}} N_{j+1,M-1}(t)$$

$$\cdot N_{j,1}(t) = \begin{cases} 1 & (t_j \leq t < t_{j+1}) \\ 0 & (\text{上記以外}) \end{cases}$$

$$\cdot Q_j = [X_j \ Y_j \ w_j] \in R^3 : \text{制御ベクトル}$$

従来の NURBS はその重心結合性のため分数係数の N 形式 B-Spline 関数を採用している。しかし前述したように、同次曲線では重心結合性が必要ないので、曲線の係数を整数化するために、この N 形式 B-Spline 関数の分母を曲線のセグメントごとにはらうことを考える（図 2 参照）。

その結果、下式のように NI 形式 B-Spline 関数 $NI_{j,M}(t)$ を定義する。

Fig.2 NI-form B-Spline function

$$\begin{cases} NI_{j,M}(t) = N_{j,M}(t) \prod_{h=2}^M C_{h,a} \quad (a: NI_{a,l}(t) = 1 \text{ となる } a \text{ の値}) \\ C_{h,a} = \prod_{b=a-h+1}^{a-1} (t_{b+h} - t_{b+1}) \quad (\text{ただし, } 0 \text{ の因子は } 1 \text{ とする}) \end{cases} \quad (5)$$

$NI_{j,M}(t)$ は加減算と乗算で表される関数なので、ノットベクトルが整数であれば、誤差のない整数演算が可能になる。ただし、パラメータに関しては有理数のままなので、点列を算出する際には誤差を含む。これを解決するためにパラメータの整数化を提案し、4.2 節に示す。

式(5)を用いることによって、平面同次化 NURBS 曲線は次式で表現される。

$$P(t) = \sum_{j=0}^{n-1} NI_{j,M}(t) Q_j \quad (6)$$

ここに、

$$P(t) = [X(t) \ Y(t) \ w(t)] \in R^3$$

$$\begin{aligned} NI_{j,M} &= (t - t_j) NI_{j,M-1}(t) \prod_{\substack{b=a-M+1 \\ b=j-1}}^{a-1} (t_{b+M} - t_{b+1}) \\ &\quad + (t_{j+M} - t) NI_{j+1,M-1}(t) \prod_{\substack{b=a-M+1 \\ b=j}}^{a-1} (t_{b+M} - t_{b+1}) \end{aligned}$$

$$\text{また, } NI_{j,l}(t) = \begin{cases} 1 & (t_j \leq t < t_{j+1}) \\ 0 & (\text{上記以外}) \end{cases}$$

$$Q_j = [X_j \ Y_j \ w_j] \in R^3: \text{制御ベクトル}$$

ここで、通常の NURBS 曲線 $p(t)$ を同次ベクトル空間で考えると ($p(t) = [x(t) \ y(t) \ 1] \in R^3$)、同次化 NURBS 曲線 $P(t)$ との間にはセグメントごとに次のような関係がある。

$$P(t) = p(t) \times \prod_{h=2}^M C_{h,a} \quad (7)$$

上式から、同次ベクトル空間において、通常の NURBS 曲線は同次化 NURBS 曲線の同値類に属することが分かる。

3.3 同次化 NURBS 曲線の幾何的性質

同次化 NURBS 曲線の持つ幾何的性質を以下に述べる。

同次化 NURBS 曲線は、同次曲線の性質と式(6)の関係から、NURBS 曲線と同様に (a)アフィン不変性 (さらに一般的な射影不変性も成立する)、(b)局所制御性、(c)変動減少性の性質を保持している。またその他の性質として (d)曲線の連続性、(e)凸包性を以下に述べる。

(d)曲線の連続性¹⁾

NURBS 曲線のセグメント間の連続性は、接続点のノットの多重度を k とすると、 C^{M-2-k} 級に保たれる³⁾。一方、同次化 NURBS 曲線はセグメント間の接続点 ($t=t_0$) において、下式が成立する。

$$P_1(t_0) = \alpha P_2(t_0) \quad (\alpha \neq 0) \quad (8)$$

一見すると、同次化 NURBS はセグメントごとに不連続のように見える。しかし、式(6)から同次化 NURBS は NURBS をある定数倍したものであり、さらに同次ベクトル空間上では同値関係が成立するので、NURBS 曲線と同様に C^{M-2-k} 級の連続性を持つ。(e)凸包性⁶⁾

2.2 節で述べたように、同次化 NURBS 曲線は任意の符号の重み係数に対して、凸包性を保つ。

4. 差分演算による高速な点列算出

NURBS は、複雑な多項式で表現されているため、曲線の形状を描画する処理が非常に重いという問題を持っている。そこで、高速に曲線上の点列を算出する差分演算³⁾を NURBS に適用することを考える。差分演算は、 n 次多項式曲線の n 階差分がパラメータ値に無関係な定数になることを利用し、加算を繰り返すことにより点列を算出する方法である。したがって、曲線上の点列を算出するために毎回漸化式を計算する方法に比べ、多項式の階差の初期値を一度算出すればあとは加算を繰り返すことにより点列を算出できる差分演算は、はるかに高速な方法であるということは明らかである。しかし、差分演算は誤差を累積しやすいため、曲線の点列算出のアルゴリズムとして用いられることはなかった。これは、浮動小数点型で表されているパラメータが繰返し演算により誤差を累積しやすいためであると考えられる。

本研究では、従来の差分演算による累積誤差の問題を解決し、高速で無誤差の点列算出アルゴリズムを構築するために、新しい基底関数を提案する。新しく提案する基底関数はパラメータを整数化し、誤差のない整数演算を可能にしたものである。

4.1 従来の差分演算

パラメータ t の多項式で表される 3 次式曲線、

$$P(t) = At^3 + Bt^2 + Ct + D \quad (0 \leq t \leq 1) \quad (9)$$

の曲線上の点は、 t の変域を n 等分し、

$$\left. \begin{aligned} t &= 0, \delta, 2\delta, \dots, n\delta \\ \delta &= 1/n \end{aligned} \right\} \quad (10)$$

なる t を式(9)に順次代入すれば計算できる。

ここで、差分演算を利用して、まったく乗算を用いることなく、高速に曲線上の点を求める方法を述べる。式(9)の (前進) 差分を計算すると次のようになる。

$$\begin{aligned} P(t) &= At^3 + Bt^2 + Ct + D \\ \Delta P(t) &= P(t+\delta) - P(t) \\ &= 3A\delta t^2 + (3A\delta^2 + 2B\delta)t + A\delta^3 + B\delta^2 + C\delta \\ \Delta^2 P(t) &= \Delta P(t+\delta) - \Delta P(t) = 6A\delta^2 t + 6A\delta^3 + 2B\delta^2 \\ \Delta^3 P(t) &= \Delta^2 P(t+\delta) - \Delta^2 P(t) = 6A\delta^3 \end{aligned}$$

第 3 階差分はパラメータ値には無関係な定数ベクトルである。一般に n 次の多項式では、第 n 階差分は定数ベクトルとなる。

$t=0, t=i\delta$ における各階差分をそれぞれ、 $P_0, \Delta P_0, \Delta^2 P_0, \Delta^3 P_0$ および $P_i, \Delta P_i, \Delta^2 P_i, \Delta^3 P_i$ と書くことにすれば、

$$\left. \begin{aligned} P_1 &= P_0 + \Delta P_0 \\ \Delta P_1 &= \Delta P_0 + \Delta^2 P_0 \\ \Delta^2 P_1 &= \Delta^2 P_0 + \Delta^3 P_0 \\ \Delta^3 P_1 &= \Delta^3 P_0 \end{aligned} \right\} \quad (11)$$

なる関係があるから、 $t=0$ における各階差分、すなわち P_0 , ΔP_0 , $\Delta^2 P_0$, $\Delta^3 P_0$ が既知であれば、 $t=\delta, 2\delta, \dots, i\delta, \dots$ における曲線上の点は、加算を繰り返すことにより順次求まるはずである。しかし、実際には例えば $\delta=1/2^3$ とした場合に、 $\Delta^3 P_0$ などは、非常に小さな値のベクトルとなり、コンピュータの語中に、十分な精度で保持することが困難となる場合もある。そこで ΔP_0 , $\Delta^2 P_0$, $\Delta^3 P_0$ をそれぞれ $1/\delta$ 倍、 $1/\delta^2$ 倍、 $1/\delta^3$ 倍した数値として格納し、式(11)の演算を行う場合には、

$$\left. \begin{aligned} P_1 &= P_0 + \delta((1/\delta)\Delta P_0) \\ (1/\delta)\Delta P_1 &= (1/\delta)\Delta P_0 + \delta((1/\delta^2)\Delta^2 P_0) \\ (1/\delta^2)\Delta^2 P_1 &= (1/\delta^2)\Delta^2 P_0 + \delta((1/\delta^3)\Delta^3 P_0) \\ (1/\delta^3)\Delta^3 P_1 &= (1/\delta^3)\Delta^3 P_0 \end{aligned} \right\}$$

とすればよい。上式で $\delta=1/2^m$ (m は正整数)とすれば、演算時間を要する乗算を用いることなく、シフト演算と加算により高速に曲線上の点を計算できる。

4.2 差分演算のNI形式への適用

NI形式同次化NURBS曲線において、差分演算を利用することを考える。まず曲線式

$$P(t) = \sum NI_{j,M}(t) Q_j \quad (12)$$

を展開し、

$$P(t) = c_{M-1}t^{M-1} + c_{M-2}t^{M-2} + \dots + c_1t^1 + c_0t^0 \quad (13)$$

の形に変形する(曲線の階数を M とする)。

パラメータ t の範囲を $t_j \leq t < t_{j+1}$ とすると、

$$\begin{aligned} t &= t_j + \frac{\delta}{n}, t_j + 2\frac{\delta}{n}, \dots, t_j + (n-1)\frac{\delta}{n} \\ \delta &= t_{j+1} - t_j \end{aligned} \quad (14)$$

である。ここで、

$$s = nt_j + i\delta \quad (i=0, 1, 2, \dots, n-1) \quad (15)$$

とおくと、

$$t = \frac{s}{n} \quad (16)$$

となる。次に同値関係を利用すると、式(13)の両辺に n^{M-1} をかけることができる。つまり、

$$\begin{aligned} \hat{P}(s) &= n^{M-1}P(t) = c_{M-1}s^{M-1} + n^1c_{M-2}s^{M-2} + \dots + n^{M-2}c_1s^1 \\ &\quad + n^{M-1}c_0s^0 \end{aligned} \quad (17)$$

のように曲線を整数化し、曲線 $\hat{P}(s)$ の k 階差分

$$\Delta^k \hat{P}(s) = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} \hat{P}(s+j\delta) \quad (18)$$

を利用して、差分演算により点列を無誤差で算出する。

しかし、式(12)から式(16)に変換するには、複雑なプログラムを組まなくてはならない。この操作を必要とする理由は、式(15)

からも明らかのように、NI形式基底関数のパラメータ値に除算が含まれるからである。

そこで、NI形式基底関数値のパラメータに関する整数化を試みる。

NI形式基底関数は、パラメータ $t (=s/n)$ を用いて、

$$\begin{aligned} NI_{j,M}(t) &= (t-t_j)NI_{j,M-1}(t) \prod_{b=j-1}^{a-1} (t_{b+M}-t_{b+1}) \\ &\quad + (t_{j+M}-t)NI_{j+1,M-1}(t) \prod_{b=j}^{a-1} (t_{b+M}-t_{b+1}) \\ NI_{j,l}(s,n) &= \begin{cases} 1 & (t_j \leq s < t_{j+1}) \\ 0 & (\text{上記以外}) \end{cases} \quad (19) \\ a : NI_{a,l}(t) &= 1 \text{ となる } a \text{ の値} \end{aligned}$$

で表される。 $NI_{j,M}(t)$ は、 $t (=s/n)$ の $M-1$ 次多項式である。

$$NI_{j,M}^2(s,n) = n^{M-1}NI_{j,M}(s/n) \quad (20)$$

とおくと $NI_{j,M}^2(s,n)^*$ は、整数化される。式(19)より $NI_{j,M}^2(s,n)$ の漸化式は、

$$\begin{aligned} NI_{j,M}^2(s,n) &= n^{M-1}(s/n-t_j)NI_{j,M-1}(s/n) \prod_{b=j-1}^{a-1} (t_{b+M}-t_{b+1}) \\ &\quad + n^{M-1}(t_{j+M}-s/n)NI_{j+1,M-1}(s/n) \prod_{b=j}^{a-1} (t_{b+M}-t_{b+1}) \\ &= n(s/n-t_j) \left\{ n^{M-2}NI_{j,M-1}(s/n) \right\} \prod_{b=j-1}^{a-1} (t_{b+M}-t_{b+1}) \\ &\quad + n(t_{j+M}-s/n) \left\{ n^{M-2}NI_{j+1,M-1}(s/n) \right\} \prod_{b=j}^{a-1} (t_{b+M}-t_{b+1}) \\ &= (s-n \cdot t_j)NI_{j,M-1}^2(s,n) \prod_{b=j-1}^{a-1} (t_{b+M}-t_{b+1}) \\ &\quad + (n \cdot t_{j+M}-s)NI_{j+1,M-1}^2(s,n) \prod_{b=j}^{a-1} (t_{b+M}-t_{b+1}) \end{aligned} \quad (21)$$

$$\begin{aligned} NI_{j,l}^2(s,n) &= \begin{cases} 1 & (nt_j \leq s < nt_{j+1}) \\ 0 & (\text{上記以外}) \end{cases} \\ a : NI_{a,l}^2(t) &= 1 \text{ となる } a \text{ の値} \end{aligned}$$

となる。ノットベクトルが整数で定義されていれば、式(15)から s は整数となる。式(21)より $NI_{j,M}^2$ 形式基底関数には、パラメータも含めて除算が含まれていないことが分かる。

同次ベクトル空間で制御ベクトルを定義し、 NI^2 形式基底関数を利用した曲線の表現式を、**NI²形式同次化NURBS曲線**と呼ぶこととする。これは、座標を同次化しただけでなくパラメータも同次化した式において、一方のパラメータを分割数 n とした式である。

従来のNI形式同次化NURBS曲線 $P(t)$ は、

* NI^2 は一つの記号として扱う。

$$P(t) = \sum NI_{j,M}(t) Q_j \quad (22)$$

である。整数化した NI^2 形式同次化 NURBS 曲線 $P(s, n)$ は、

$$\begin{aligned} P(s, n) &= \sum NI_{j,M}^2(s, n) Q_j \\ &= \sum n^{M-1} NI_{j,M}(s/n) Q_j \\ &= n^{M-1} \sum NI_{j,M}(t) Q_j \\ &= n^{M-1} P(t) \end{aligned} \quad (23)$$

で表せる。よって $P(t)$ と $P(s, n)$ は同値関係にある。また、式(17)と式(4.15)の比較により、

$$P(s, n) = \hat{P}(s) \quad (24)$$

したがって、 $P(s, n)$ の k 階差分は

$$\Delta^k P(s, n) = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} P(s + j\delta, n)$$

で求められる。以上から、基底関数を式(21)のように変形することによって完全整数演算による無誤差の点列算出を行うことができることがわかる。

4.3 差分演算による点列算出実験

差分演算による同次化 NURBS 曲線の点列算出がどれだけ優れているかを調べるために累積誤差と算出速度の両面で実験を行う。

まず、従来の差分演算で問題であった累積誤差が NI^2 形式を用いることによってどのように回避されるかを(1) **曲線の終点における累積誤差**の実験で調べる。実験対象は N 形式同次化 NURBS 曲線と NI^2 形式同次化 NURBS 曲線とする。前者のデータが従来の NURBS のデータと見なすことができる(NURBS の場合、実際にはさらに除算を行う必要がある)。

次に、(2) **曲線上の点列算出速度**の実験を行う。この実験は次に示す4種類の場合を行う。

- N 形式同次化 NURBS において de Boor Cox の漸化式を利用し点列算出したもの
 - N 形式同次化 NURBS 曲線を多項式に変形してパラメータを代入したもの
 - NI 形式同次化 NURBS 曲線を多項式に変形しホーナー則を利用したもの
 - NI^2 形式同次化 NURBS 曲線に差分演算を利用したもの
- ただし b), c) の実験では基底関数から多項式を求める時間は含まれていない。d) では前進差分を求める時間を含んでいる。また初期計算として、b), c) のために多項式に変形する時間と、d) のために各階差分を求める時間の比較が必要になる。そこで初期計算にかかる時間の比較も行う。さらに、点列を算出するために必要な演算回数の比較も行う。

これらの実験に利用した曲線データを以下に示す。

3 階の場合の制御点

$$\begin{bmatrix} 50 & 400 & 100 & 1 \end{bmatrix}, \begin{bmatrix} 100 & 100 & 100 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 300 & 300 & 100 & 1 \end{bmatrix}$$

4 階の場合の制御点

$$\begin{bmatrix} 50 & 400 & 100 & 1 \end{bmatrix}, \begin{bmatrix} 100 & 100 & 100 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 300 & 300 & 100 & 1 \end{bmatrix}, \begin{bmatrix} 500 & 50 & 100 & 1 \end{bmatrix}$$

5 階の場合の制御点

$$\begin{bmatrix} 50 & 400 & 100 & 1 \end{bmatrix}, \begin{bmatrix} 100 & 100 & 100 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 300 & 300 & 100 & 1 \end{bmatrix}, \begin{bmatrix} 500 & 50 & 100 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 600 & 400 & 100 & 1 \end{bmatrix}$$

ノットはすべてユニフォームなものとする。

(1) 曲線の終点における累積誤差

N 形式同次化 NURBS 曲線と、 NI^2 形式同次化 NURBS 曲線に差分演算を適用した場合の累積誤差の実験を行う。実験では、セグメント数 1, 階数 3, 4 の曲線に対して、点列算出数を 200~1000 に変化させた場合における 2 つの方法の実験結果を比較する。実験結果を表 1 に示す。

Table 1 The accumulated error at the end point

		200	600	1000
3rd order	N form	200.00000000049	200.00000000248	199.99999999142
	accumulated error	0.00000000049	0.00000000248	0.00000000858
	NI^2 form	200.00000000000	200.00000000000	200.00000000000
4th order	N form	300.00000023652	299.99999583742	300.00002271778
	accumulated error	0.00000023652	0.00000416258	0.00002271778
	NI^2 form	300.00000000000	300.00000000000	300.00000000000

実験結果から分かるように、N 形式は算出する点列数の増加に伴って誤差が大きくなっている。一方、 NI^2 形式は整数演算であるため、1 回の割り算による誤差のみで(ユークリッド座標に変換する前は誤差がない)、算出する点列数に影響されない。実験の結果、 NI^2 形式同次化 NURBS 曲線は無誤差演算が可能となり、N 形式同次化 NURBS 曲線は累積誤差を生じることが分かる。N 形式の曲線に拡張差分演算を適用してもなお累積誤差を生じるのは、N 形式の多項式の係数に誤差が含まれているためである。

(2) 曲線上の点列算出速度

上に示した 4 種類の曲線について、点列の算出にかかる時間の比較実験の結果を図 3 に示す。ただし、この実験に用いた曲線は 3 階である。

Fig.3 Points computation time (3 階)

図 3 から、従来の漸化式による方法より拡張差分演算を用いた方が速度面において明らかに優れていることが分かる。このような結果の理由は、de Boor Cox の漸化式を用いる従来の方法が漸化式による関数値の算出と制御ベクトルとの線形結合を点列数の回数分繰返さなくてはならないのに対し、拡張差分演算を用いる方法は最初に 1 回のみ各階差データを求めた後は、乗除算より高速に計算できる加減算で構成される差分演算を点列数の回数分繰返すだけだからである。またホーナー則を利用した場合と比較しても、拡張差分演算の方が演算回数が少なく算出時間が短くなることが分かる。また注意したいのが、この実験において b) N 形式多項式と c) NI 形式ホーナー則では漸化式から多項式への変形にかかる時間は除いており、d) NI^2 形式に差分演算を利用したものでは初期階差の計算時間を含んでいることである。参考とし

て、表 2 にそれぞれの初期値の計算にかかる時間を比較する。この実験では X 座標のみの計算を行っている。

Table 2 Calculation time for initial value (μ s)

N form	difference	120	210	330
	polynomial	1202	1963	2814
NI ² form	difference	150	270	430
	polynomial	1181	1933	2794

差分演算では、初期値である各階差分を求めるために、2 から 3 点の点列は de Boor Cox の漸化式から直接計算している。一方、多項式は基底関数を漸化式から計算し、曲線式を多項式の形に変形するために要する時間を示している。表から明らかなように、初期値の計算を考えても、差分演算を用いる方が優れていることが分かる。

次に、点列を算出する際に必要となる演算の回数を比較してみる。一般的に M 階の曲線においては、表 3 に示すようになる。ただし、これには初期計算にかかる時間は除いている。また、ユークリッド座標値を求めるための射影にともなう 1 回の除算は含めていない。図 3 の結果は以下の演算回数の比較からも、容易に予想される結果である。

Table 3 The calculation times (for one coordinate)

	+, −	÷, ×
polynomial	$M-1$	$M(M-1)/2$
Horner's method	$M-1$	$M-1$
Differential operation	M	0

5. 結 論

以下に本研究の結論をまとめる。

- (1) 有理式表現された従来の NURBS を非有理表現し、さら

に NI² 形式 B-Spline 関数の提案により、加減算と乗算に基づいて算出される NI² 形式同次化 NURBS を提案した。

- (2) 同次化 NURBS は非有理式表現されているため数値的に安定で、また従来の NURBS では表現できなかった無限遠点を表現することができる。
- (3) 同次化 NURBS は同次表現されているため、同次ベクトル空間において重み係数の正負にかかわらず凸包性を有する。
- (4) NI² 形式同次化 NURBS に拡張差分演算を適用することにより、曲線上の点を連続的に、無誤差でかつ高速に算出することができる。

参 考 文 献

- 1) M.E.Hohmeyer and B.A.Barsky : Rational Continuity : Parametric, Geometric, and Frenet Frame Continuity of Rational Curves, ACM Trans. Graphics 8, 4,(1989).
- 2) L. Piegle and W. Tiller : The NURBS Book, Springer-Verlag, (1995) .
- 3) 山口富士夫：コンピュータディスプレイによる形状処理工学Ⅰ，日刊工業新聞社，(1982) .
- 4) 山口富士夫：コンピュータディスプレイによる形状処理工学Ⅱ，日刊工業新聞社，(1982) .
- 5) 山口富士夫：4次元理論による図形・形状処理工学，日刊工業新聞社，(1996) .
- 6) 山田 敦，松村 潔，山口富士夫：同次幾何的ニュートン法による有理曲線，曲面に対する干渉処理，精密工学会誌，61, 2 (1995) 203.
- 7) 山田 敦，松村 潔，新関雅俊，山口富士夫：曲線・曲面に対する 4・4 行列式法の適用（第 3 報）－符号付き同次座標で定義された有理曲線に対する干渉処理－，精密工学会誌，60, 2 (1994) 111.