スプライン曲線とベジエ曲線の混合描画のユーザインタフェース

箕　原　辰　夫

1．はじめに

ベクトル2次元描画において，一番使われているソフトウェアは，Adobe社のIllustrator[1]である。このAdobe Illustratorを始めとして，ほとんどの2次元ベクトル描画ソフトウェアや2次元グラフィックスのライブラリにおいては，ベジエ（Bézier）曲線が基本的な描画手法として使われている。しかしながら，3次元のモデリングにおいては，正確なモデリングが要求されるCAD系のソフトウェアを中心に，非一様有利Bスプライン（Non-Uniform Rational B-Spline: NURBS）曲線が使われていることが多い。造形手法的にも，NURBS曲線からそのまま造形をしてゆくもの（たとえば，solidThinking[2]やRhinoceros 3D[3]など）や，NURBS曲面用の制御点（直線から構成されるネット形状上の接点）から，副分割して構成する曲面（Subdivision Surface: SDS）により造形をするもの（Blender[4]やAutodesk Maya[5]など）と分かれているが，どちらにせよ，造形の中心はBスプライン曲線が基本となっている。もちろん，3次元モデリングソフトウェアにおいても，ベジエ曲線の描画も採り入れている場合も珍しくはない。ただ，2次元のベクトル描画ソフトウェアにおいて，NURBS曲線による描画ではなく，ベジエ曲線による描画手法のみが用いられてきた背景には，Adobe Illustratorからの踏襲という歴史的な要因が強いと思われる。

数学的には，既にベジエ曲線とBスプライン曲線は同一の曲線の表現方法を変えただけであるので，相互に変換する方法が研究されているし，また3次元については，有理ベジエ曲面と有理スプライン曲面を変換することも可能となっている。そのため，まず2次元描画においても，NURBS曲線の手法を用いて，描画することには技術的な問題はないと思われる。実際に，3次元モデリングにおいても，NURBS曲線を最初に描くときは，2次元的な平面に描画を起こすことが多い。本論文では，ベジエ曲線とスプライン曲線，およびBスプライン曲線のそれぞれの描画手法を選択し，1つのモデリングの中で混在して

(1) Adobe Illustrator CS5ユーザガイド，Adobe Systems Incorporated, 500pp, 2010，日本語オンライン版 http://help.adobe.com/ja_JP/illustrator/cs/using/index.html

(2) solidThinking User Manual, solidThinking and Altair Inc., http://www.solidthinking.com/Navigation.aspx?top_nav_name=Support&item_name=tutorials，2011年12月閲覧.

(3) Rhinoceros Training Text Level1 and Level2 version 4.0, Robert McNeel & Associates, 2007，日本語版 http://www.rhino3d.co.jp/support/download.html，2012年1月閲覧.

(4) Blender User's Documentation, blender.org, http://wiki.blender.org/index.php/Main_Page，2012年1月閲覧.

(5) Autodesk Maya 2011 PDF Documents, Autodesk, 2011，日本語版 http://www.autodesk.co.jp/adsk/servlet/item?siteID=1169823&id=14937639&linkID=14165148, 2012年1月閲覧.

いきながら描画できるソフトウェアのインタフェースを構築し，そのための理論的な基礎を整備し，また実際にプロトタイプのソフトウェアを試作したので，それらを報告する。

また，先駆をなして，他の2次元ベクトル描画ソフトウェアに多大な影響を及ぼしたAdobe Illustratorの描画手法（ペンツールによる曲線の描画）では，ベジエ曲線を本来の形で描画するのでなく，曲線の方向をしめすような形で描画が行なわれる。これは，複数の曲線から構成される複合曲線をC^2接続（制御点での2階微分形が一致する接続）で連続した形（Adobe Illustratorではスムーズコーナーと呼んでいる）で描くために考案されたのではないかと推測される。曲線の方向（2つの端点の接線ベクトル）によって描画する曲線は，本来数学的にはエルミート（Hermite）曲線・補間が該当する。以下の章で示すように，エルミート曲線は，ベジエ曲線に変換・吸収することが可能である。そのため，Adobe Illustratorではエルミート曲線のような描画方法でベジエ曲線を描いて行く仕様になっている。しかしながら，本来のベジエ曲線として，端点と補助点から描画することも可能である。今回の試作実装では，ベジエ曲線の描画に，これまで行なわれてきたエルミート曲線風の描画手法と，本来のベジエ曲線のための描画手法を分けて，実現することにした。

2．曲線の変換

曲線の理論的な基礎としては，3次の多項式曲線を中心に解説しているRogersの教科書[6]や一般的に普及しているFoleyの教科書[7]があり，それらはわかりやすいが，ここでは，3次よりも高次の多項式曲線を扱う。ここでの表記は，Farin[8][9]の教科書や穂坂の教科書[10]に高次の多項式曲線の問題が掲載されているため，それらの教科書の表記法に基づいている。

2.1．エルミート曲線（補間）とベジエ曲線の間の変換

エルミート曲線は，2つの端点とその点における接線ベクトル（tangent vector）から，曲線を構成する。3次以上の高次の場合は，5次の場合[11]になるが，この場合は，接線ベクトルだけでなくて，2階微分ベクトルの情報を用いる。それ以上の$2n+1$次のエルミート多項式曲線においては，n階微分ベクトルの情報を用いる。しかしながら，一般的に描

---

(6) David F. Rogers and J. Alan Adams, Mathematical Elements for Computer Graphics, Second Edition, McGraw-Hill, 1990, 邦訳『コンピュータグラフィックス第2版』，川合慧監訳，日刊工業新聞社，1993.

(7) James D. Foley, Computer graphics: principles and practice, Addison-Wesley Professional, 1996, 邦訳『コンピュータグラフィックス：理論と実践』，佐藤義雄監訳，オーム社，2001.

(8) Gerald E. Farin, Curves and Surfaces for CAGD: A Practical Guide, fifth edition, Series in Computer Graphics and Geometric Modeling, Morgan Kaufmann, 2002.（以下CAGDで略記する）

(9) Gerald E. Farin, NURBS: From Projective Geometry to Practical Use, second edition, A K Peters, 1999, 邦訳『NURBS－射影幾何学から実務まで－』第2版，原孝成他訳，共立出版，2001.

(10) Mamoru Hosaka, Modeling of Curves and Surfaces in CAD/CAM, Springer-Verlag, 1992, 日本語版　穂坂衛，『CAD/CAMにおける曲線曲面のモデリング』，東京電機出版局，1996.

(11) Farin, CAGD, pp.106-107.

画ソフトウェアで接線ベクトル以上のベクトル情報を表示することはないだろう。高次の曲線をエルミート曲線に変換するときは，複合曲線に分解し，分解された各曲線セグメントを 3 次のエルミート曲線で近似するしかない。ここでは，3 次のベジエ曲線とエルミート曲線の間の変換を扱う。

1 つのベジエ曲線 $\mathbf{b}(t)$ は，媒介変数 t を用いたパラメトリック形式で表現されるが，n 次のバーンスタイン（Bernstein）多項式 B_i^n を用いて次のように表される。通常は，n は 2 以上と仮定される。

$$\mathbf{b}(t) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(t) \tag{1}$$

この式 1 において，$\mathbf{b}_i$ は，ベジエ曲線の凸包（convex hull）を構成する制御多角形の点である。$\mathbf{b}_i$ はアフィン空間上の位置ベクトルを示す。バーンスタイン多項式 B_i^n は，バーンスタイン関数とも呼ばれるが，次のように定義される。

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{2}$$

ここで 2 項係数は次のように定義される。

$$\binom{n}{i} = \begin{cases} \dfrac{n!}{i!(n-i)!} & if \quad 0 \le i \le n \\ 0 & else \end{cases} \tag{3}$$

図 1 は 3 次のベジエ曲線の制御多角形の例と制御多角形上の制御点を示している。

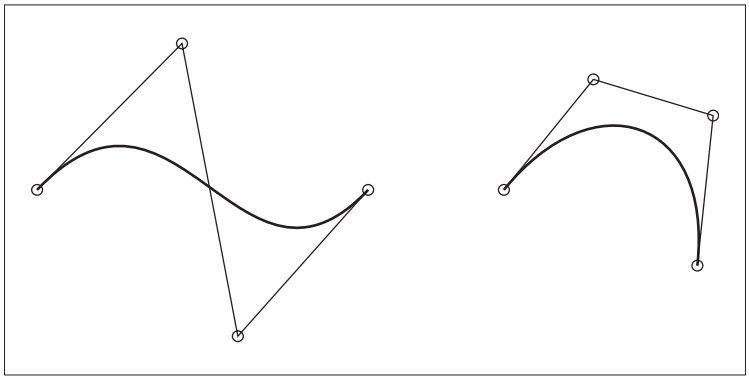

図 1　3 次のベジエ曲線の制御多角形の例

さて，ベジエ曲線とエルミート曲線の関係を求めるためには，まず，ベジエ曲線の導関数 $\mathbf{b}'(t)$ が必要になり，以下の通りとなる。

$$\mathbf{b}'(t) = n \sum_{i=0}^{n-1} (\mathbf{b}_{i+1} - \mathbf{b}_i) B_i^{n-1}(t) \tag{4}$$

3次のエルミート曲線 $\mathbf{p}(t)$ は，図2に示すように，2つの端点（$\mathbf{p}_0$ および $\mathbf{p}_1$）とそれぞれの端点における接線ベクトル（$\mathbf{m}_0$ および $\mathbf{m}_1$）から表現される。媒介変数 t の範囲を次のように対応させる。

$$\begin{aligned} \mathbf{p}(0) &= \mathbf{p}_0, \quad \acute{\mathbf{p}}(0) = \mathbf{m}_0 \\ \mathbf{p}(1) &= \mathbf{p}_1, \quad \acute{\mathbf{p}}(1) = \mathbf{m}_1 \end{aligned} \tag{5}$$

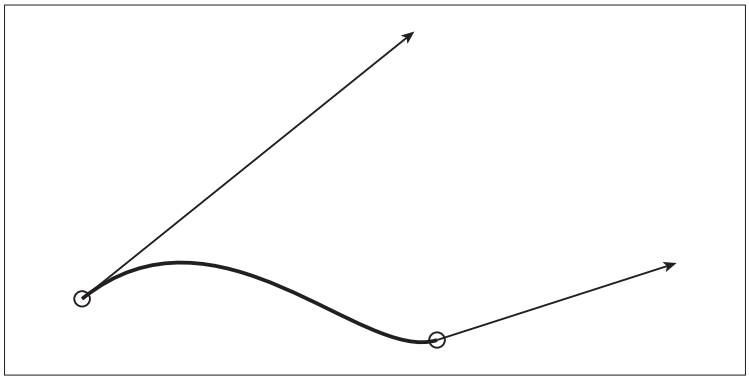

図2　3次のエルミート曲線の例

次のように端点をベジエ曲線の制御（$\mathbf{b}_0$ および $\mathbf{b}_3$）に対応させ，3次のベジエ曲線を仮定し，（式4）の形に当てはめる。

$$\mathbf{b}_0 = \mathbf{p}_0, \quad \mathbf{b}_3 = \mathbf{p}_1 \tag{6}$$

3次のベジエ曲線の残された2つの中間の制御点（$\mathbf{b}_1$ および $\mathbf{b}_2$）とエルミート曲線の接線ベクトルの関係は，（式4）から以下のようになる[12]。

$$\mathbf{b}_1 = \mathbf{p}_0 + \frac{1}{3}\mathbf{m}_0, \quad \mathbf{b}_2 = \mathbf{p}_1 - \frac{1}{3}\mathbf{m}_1 \tag{7}$$

上記からみてわかる通り，エルミート曲線を使った場合，中間の制御点の1つ目は接線ベクトルの方向，2つ目は接線ベクトルの反対方向に1/3の大きさに伸ばした場所になっている。Adobe Illustrator など，ほとんど2次元ベクトル描画ソフトウェアでは，反対方向に同じ大きさの接線ベクトルが出て，それが中間の制御点になっている。これは先に述べたように，複合曲線において，曲線間の C^2 連続性を保証するためであり，厳密にはエルミート曲線の描画方法とは異なる。本論文では，Adobe Illustrator などの描画方法を，「エルミート曲線風の描画方法」と称している。

逆に3次のベジエ曲線からエルミート曲線に変換する際は，まず，次のようにエルミート曲線の多項式を記述する。

---

[12] Farin, CAGD, p.103.

$$\mathbf{p}(t) = \mathbf{p}_0 H_0^3(t) + \mathbf{m}_0 H_1^3(t) + \mathbf{m}_1 H_2^3(t) + \mathbf{p}_1 H_3^3(t) \tag{8}$$

それぞれの項の係数は，バーンスタイン多項式を用いて，以下のように表される。

$$\begin{aligned} H_0^3(t) &= B_0^3(t) + B_1^3(t), \quad & H_1^3(t) &= \frac{1}{3} B_1^3(t), \\ H_2^3(t) &= \frac{1}{3} B_2^3(t), & H_3^3(t) &= B_2^3(t) + B_3^3(t) \end{aligned} \tag{9}$$

2.2. スプライン曲線とBスプライン曲線の間の変換

Maya なども含めて，曲線上の制御点を指定していき，制御点に従って滑らかな曲線を補間してゆくような形で曲線描画ができるソフトウェアがある。このような曲線と補間は，スプライン曲線あるいはスプライン補間と呼ばれているが，より一般には多項式補間（polynomial interpolation）と呼ばれている。多項式補間の方法において，ラグランジュ補間は，複数の制御点から構成される場合，端点の近くにおいて発振する現象[13]があり，あまり用いられない。ニュートン補間においては，丸めなどで精度が落ちたときに，発振しないという証明がなされていない[14]。エルミート補間は，曲線上の各制御点における接線ベクトルが与えられた条件のときに，主に 3 次で補間してゆく形の限られたものである。一般的にはBスプライン多項式を用いた表現形式の形で多項式を表現し，補間操作を行なっている。ここでは，Bスプライン曲線およびBスプライン多項式を説明し，Bスプライン多項式を用いて，補間操作として C^2 連続性が保障されたスプライン曲線を構成してゆく方法について説明する。

n 次のBスプライン曲線（basis spline）は，連続した複合曲線から構成される。個々の曲線をセグメント（segment）と呼ぶ。媒介変数を使って曲線セグメントの区切りを表す点をノット（knot）と呼び，ノットの列を使って，セグメントがどのように区切られているかを示す。また，凸包となる制御多角形を構成する点を用いても曲線が表現される。数式的に定義すると，非有理Bスプライン曲線は，次の項目から定義できる。

・曲線の次数 n
・曲線のセグメント数 K
・$K+1$ 個のノット列 $u_0, \cdots u_K$　ただし，$u_i \leq u_{i+1}$
・制御多角形の頂点 $\mathbf{d}_0, \cdots \mathbf{d}_L$　ただし，$L = K-n+1$

一様Bスプライン曲線は，ノット列の間隔が一様になっていて，整数値で表されることが多い。ただし，ノット自体は重複してもよい。これらの項目を使って，Bスプライン曲線 $\mathbf{s}(u)$ は次のように定義される。

$$\mathbf{s}(u) = \sum_{i=0}^{L} \mathbf{d}_i N_{i+1}^n(u) \tag{10}$$

[13] Farin, CAGD, p.101.
[14] Farin, CAGD, p.116.

ここで，Bスプライン多項式 $N_i^n(u)$ は，Bスプライン基底関数とも呼ばれるが，Mansfield と de Boor と Cox の帰納式によって，次のように定義される[15]。

$$N_i^n(u) = \frac{u-u_{i-1}}{u_{i+n-1}-u_{i-1}}N_i^{n-1}(u)+\frac{u_{i+n}-u}{u_{i+n}-u_i}N_{i+1}^{n-1}(u) \tag{11}$$

$$N_i^0 = \begin{cases} 1 & if\ u_{i-1} \le u < u_i \\ 0 & else \end{cases} \tag{12}$$

3次のBスプライン曲線について，ノット列に重複のある形のものを図3に記した。端点を構成するためには，3次の場合には，ノットに3重の重複がなければならないのがわかる。

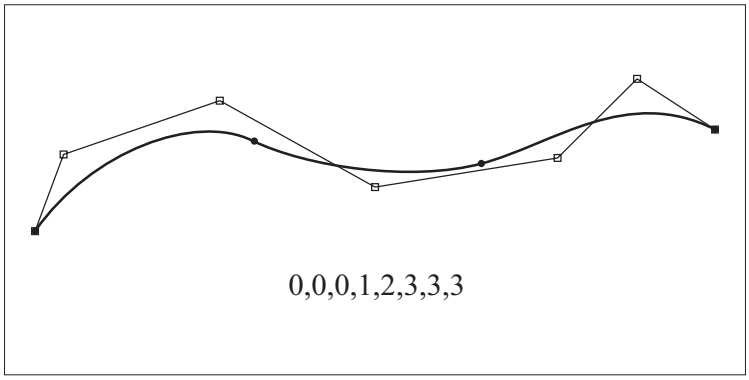

図3　3次のBスプライン曲線の例

C^2 連続性が保障されたスプライン曲線を，3次のBスプライン曲線を用いて補間して表現する。より高次のBスプライン曲線を用いて補間することも可能であるが，最低限，3次のBスプライン曲線で C^2 連続性が保障されるからである。まずノット列 ν_i が与えられると仮定するが，端点を構成するためには，3次の場合，3重にする必要がある。形式的に与えられるものを記述すると，次のようになる。

・曲線上の点の列：$\mathbf{p}_0, \cdots \mathbf{p}_K$
・ノット列：$\nu_0, \cdots \nu_K$　ただし，ν_0 および ν_K は，3重であるとする。

このときBスプライン曲線を構成する制御多角形の頂点の列 $\mathbf{d}_0, \cdots \mathbf{d}_L$　ただし $L = K-n+1$，を求め，C^2 連続性が保障されたスプライン曲線 $\mathbf{x}(u)$ を求める。なお，この $\mathbf{x}(u)$ は，次の式を満足する。

$$\mathbf{x}(\nu_i) = \mathbf{p}_i \quad ここで \quad i = 0, \cdots K \tag{13}$$

以下に3次Bスプライン曲線で補間された例を示す。

[15] Farin, CGAD, pp.142-144.

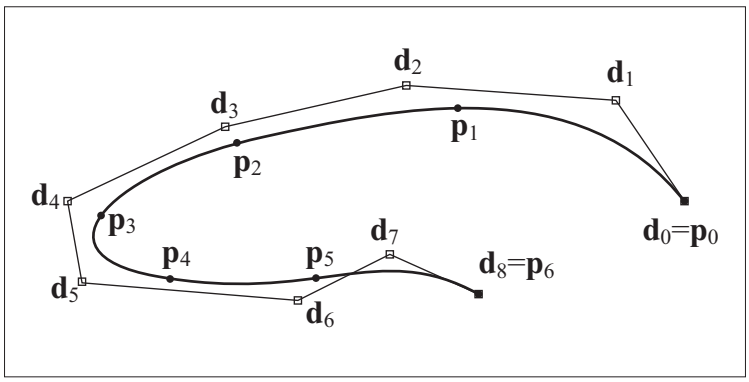

図 4　補間した 3 次のＢスプライン曲線の例

解の一意性を保障するため，端点固定条件（clamped end conditions）として，２つの端点におけるスプライン曲線の一階微分形（接線ベクトル）を次のように仮定する。

$$\dot{\mathbf{x}}(\nu_0) = \frac{3}{\nu_1 - \nu_0}[\mathbf{d}_1 - \mathbf{d}_0], \quad \dot{\mathbf{x}}(\nu_K) = \frac{3}{\nu_K - \nu_{K-1}}[\mathbf{d}_L - \mathbf{d}_{L-1}] \tag{14}$$

まず，３重ノットの端点が保証されることから，ただちに次の解が得られる。

$$\mathbf{d}_0 = \mathbf{p}_0, \ \mathbf{d}_L = \mathbf{p}_K \tag{15}$$

次に，端点固定条件（式13）を展開することによって，次の２つの制御多角形の頂点も求められる。

$$\mathbf{d}_1 = \mathbf{d}_0 + \frac{\nu_1 - \nu_0}{3}\dot{\mathbf{x}}(\nu_0), \quad \mathbf{d}_{L-1} = \mathbf{d}_L + \frac{\nu_K - \nu_{K-1}}{3}\dot{\mathbf{x}}(\nu_K) \tag{16}$$

残りの制御多角形の頂点は，次のような３次のＢスプライン曲線の性質を満たすことになる。

$$\mathbf{p}_i = \mathbf{d}_i N_i^3(\nu_i) + \mathbf{d}_{i+1} N_{i+1}^3(\nu_i) + \mathbf{d}_{i+2} N_{i+2}^3(\nu_i) \quad \text{ここで } i = 2, \cdots K-2 \tag{17}$$

特に，ノット列が一様である場合（$\nu_i = i$），（式17）は，次のように展開できる。

$$6\mathbf{p}_i = \mathbf{d}_i + 4\mathbf{d}_{i+1} + \mathbf{d}_{i+2} \quad \text{ここで } i = 2, \cdots K-2 \tag{18}$$

（式18）について，連立一次方程式の解を求めれば，すぐに頂点列 $\mathbf{d}_2, \cdots \mathbf{d}_{L-2}$ を求めることが可能になる。

2.3. ベジエ曲線とＢスプライン曲線の変換

ベジエ曲線をＢスプライン曲線に変換するのは，もともとＢスプライン曲線がベジエ曲線の超集合になっているので，Ｂスプライン曲線をベジエ曲線に合わせた形で，ベジエ曲線の制御多角形の頂点に合わせる形で，該当するノットを多重定義するだけで良い。n 次のベジエ曲線の場合，Ｂスプラインのノットを，n 重にするだけで，端点を表せる。本数 r のベジエ曲線が C^1 接続（１階微分形が連続された形）で繋ぎ合わされている場合は，途中の $r-1$ 個の端点は $n-1$ 重にするだけでよい。図５は，３次のベジエ曲線を３次のスプライン曲線に変換したものである。

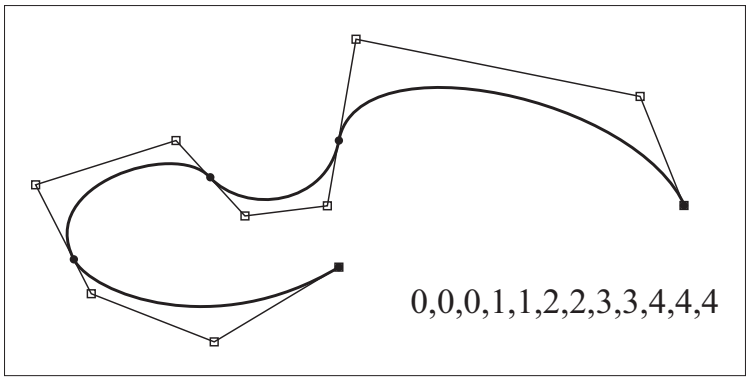

図５　３次のベジエ曲線を３次のＢスプライン曲線に変換

逆に，任意のＢスプライン曲線を C^2 接続されたベジエ曲線の複合曲線に変換するためには，Ｂスプライン曲線の特定の区間上の制御多角形の頂点（ブロッサム：blossom と呼ぶ）に，ド・ブール（de Boor）のアルゴリズムを適用する必要がある。ド・ブールのアルゴリズム[16]は，ベジエ曲線におけるド・カステリョ（de Casteljau）のアルゴリズム[17]の一般形になっている。

まずブロッサムの定義[18]であるが，ブロッサムは，多項式に適用できる関数である。制御多角形の頂点が図６のように，$\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2$ の３つがあるとすると，これをブロッサムの形式で記述すると次のようになる。

$$\mathbf{b}[0,0] = \mathbf{b}_0,\ \mathbf{b}[0,1] = \mathbf{b}_1,\ \mathbf{b}[1,1] = \mathbf{b}_2 \tag{19}$$

次に，図６の各辺上を自由に動く２点があるとする。それぞれの位置を表すために，媒介変数 t と $s(0 \leq t \leq 1,\ 0 \leq s \leq 1)$ を用いる。この変数を用いて，図６の各辺上にある白い点は，それぞれ次のようにブロッサムで表される。

---

[16] De Boor's Algorithm, Wikipedia, http://en.wikipedia.org/wiki/De_Boor's_algorithm

[17] De Casteljau's Algorithm, Wikipedia, http://en.wikipedia.org/wiki/De_Casteljau's_algorithm

[18] Farin, CAGD, pp.30-34.

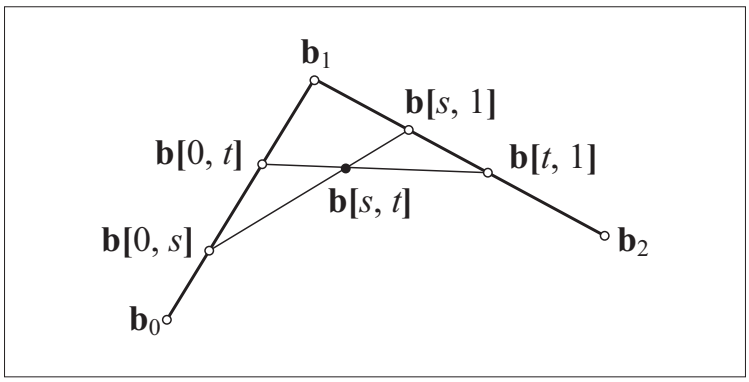

図 6　ブロッサムの記法（メネラオスの定理）

$$\begin{aligned}
\mathbf{b}[0,t] &= (1-t)\mathbf{b}_0+t\mathbf{b}_1 &&= (1-t)\mathbf{b}[0,0]+t\,\mathbf{b}[0,1]\\
\mathbf{b}[0,s] &= (1-s)\mathbf{b}_0+s\mathbf{b}_1 &&= (1-s)\mathbf{b}[0,0]+s\mathbf{b}[0,1]\\
\mathbf{b}[t,1] &= (1-t)\mathbf{b}_1+t\mathbf{b}_2 &&= (1-t)\mathbf{b}[0,1]+t\,\mathbf{b}[1,1]\\
\mathbf{b}[s,1] &= (1-s)\mathbf{b}_1+s\mathbf{b}_2 &&= (1-s)\mathbf{b}[0,1]+s\mathbf{b}[1,1]
\end{aligned} \tag{20}$$

ここから，更にこれらの点を結んでできる次のような 2 つの交点をブロッサムで記述することができる。

$$\begin{aligned}
\mathbf{b}[s,t] &= (1-t)\mathbf{b}[s,0]+t\mathbf{b}[s,1]\\
\mathbf{b}[t,s] &= (1-s)\mathbf{b}[0,t]+s\mathbf{b}[t,1]
\end{aligned} \tag{21}$$

メネラオスの定理[19]から，この 2 つの交点は等しいことがわかっている。

$$\mathbf{b}[s,t] = \mathbf{b}[t,s] \tag{22}$$

この交点 $\mathbf{b}[s,t]$ において，s と t をそれぞれ定義域内で変化させた結果は，2 次のベジエ曲線 $\mathbf{b}^2(t)$ と等しい。このブロッサムの記法では，2 つの媒介変数を用いたが，変数の数は任意である。いま，n 個の変数を用いたブロッサムがあるとすると，以下の対称性（symmetry）が成り立つ。

$$\mathbf{b}[t_1,\cdots t_n] = \mathbf{b}[\pi(t_1,\cdots t_n)] \tag{23}$$

ここで，$\pi(t_1,\cdots t_n)$ は，媒介変数の順序 $t_1,\cdots t_n$ を入れ替えたもの（permutation）である。たとえば，$\mathbf{b}[0,t] = \mathbf{b}[t,0]$，$\mathbf{b}[t_1,t_2,t_3] = \mathbf{b}[t_2,t_3,t_1]$ などが言える。またブロッサムは，多線形（multilinear）である。

$$\mathbf{b}[(1-\alpha)t+\alpha s,\cdots] = (1-\alpha)\mathbf{b}[t,\cdots]+\alpha\mathbf{b}[s,\cdots] \tag{24}$$

---

[19] Menelaus’ theorem, Wikipedia, http://en.wikipedia.org/wiki/Menelaus%27_theorem

また，ブロッサムのすべての媒介変数が同一であれば，それは多項式曲線を示すことになるが，いくつかの変数が同一の場合，次のような記法を用いる。以下の式において，変数 t は n 回重複して現れている。

$$\mathbf{b}[t, \cdots t] = \mathbf{b}[t^{\langle n \rangle}] = \mathbf{b}(t) \tag{25}$$

ベジエ曲線における制御点は，n 個の変数を用いたブロッサムに次のように記述される。

$$\mathbf{b}_i = \mathbf{b}[0^{\langle n-1 \rangle}, 1^{\langle i \rangle}] \tag{26}$$

たとえば，3 次ベジエ曲線において，$\mathbf{b}_0 = \mathbf{b}[0^{\langle 3 \rangle}, 1^{\langle 0 \rangle}] = \mathbf{b}[0, 0, 0]$ である。ただし，ベジエ曲線が，区間 $[0, 1]$ ではなく，任意の区間 $[a, b]$ 上で定義されている場合は，次のようになる。

$$\mathbf{b}_i = \mathbf{b}[a^{\langle n-1 \rangle}, b^{\langle \mathrm{i} \rangle}] \tag{27}$$

次に，ド・カステリョのアルゴリズムにより，n 個の変数を用いたブロッサムを次のように展開することができる。

$$\begin{aligned} \mathbf{b}[0^{\langle n-r-i \rangle}, t^{\langle r \rangle}, 1^{\langle i \rangle}] &= (1-t)\mathbf{b}[0^{\langle n-r-i+1 \rangle}, t^{\langle r-1 \rangle}, 1^{\langle i \rangle}] \\ &\quad + t\mathbf{b}[0^{\langle n-r-i \rangle}, t^{\langle r-1 \rangle}, 1^{\langle i+1 \rangle}] \end{aligned} \tag{28}$$

任意の区間 $[a, b]$ 上で定義されている場合は，上記の 0 と 1 の部分には，a と b で置換される。

n 次のＢスプライン曲線において，単調増加するノット列を $u_0, \cdots u_k$ と置いたとき，この曲線は，$p = K-n+1$ の制御多角形 $\mathbf{P}$ によって定義される。つまり，連続した n 個のノットからなる組の数と同じだけの制御点がある。

$$\mathbf{P} = \mathbf{d}_0, \mathbf{d}_1, \cdots \mathbf{d}_p \tag{29}$$

n 個の連続した区間はすべて，制御多角形の 1 辺に射影される。

$$[u_i, u_{i+n}] \rightarrow \mathbf{d}_i \mathbf{d}_{i+1} \tag{30}$$

ここで，区間 $\lambda = [u_I, u_{I+1}]$ を 2 つ連続するノット u_I, u_{I+1} に定義する。n 個の連続区間の一部である区間 λ について考えると，ノット列のすべての区間を見る必要はない。$n-1 \leq I \leq K-n$ が成り立つ。その区間 λ を定義域間隔（domain interval）[20] と呼ぶ。その区間の制御多角形と制御点を，それぞれ $\mathbf{P}^{\lambda}, \mathbf{d}^{\lambda}$ と表現する。

---

[20] Farin, NURBS, 邦訳 p.160.

$$\mathbf{P}^{\lambda} = \mathbf{d}_0^{\lambda}, \cdots \mathbf{d}_n^{\lambda} = \mathbf{d}_{I-n+1}, \cdots \mathbf{d}_{I+1} \tag{31}$$

$u \in \lambda$ とすると，u は制御多角形の辺上の点としても射影されるので，線形補完として次のように記述することができる。

$$\mathbf{d}_i^1(u) = \frac{u-u_I}{u_{I+1}-u_I}\mathbf{d}_{i-1} + \frac{u_I-u}{u_{I+1}-u_I}\mathbf{d}_i \tag{32}$$

このようにノット列に 1 つのノット u を追加すると，制御多角形 $\mathbf{P}^{\lambda}$ は新しい多角形 $\mathbf{P}[u]$ に変わる。

$$\mathbf{P}[u] = \mathbf{d}_0, \mathbf{d}_0^1(u), \cdots \mathbf{d}_n^1(u), \mathbf{d}_n \tag{33}$$

この過程は，W. Boehm によりノット挿入（knot insertion）と名付けられている。そこで，n 回ノット u を追加した場合，ド・ブーアのアルゴリズムと呼ばれている以下の再帰式によって，追加された制御点 $\mathbf{d}_i^r(u);\ r = 1, \cdots n;\ i = r, \cdots n$ を線形補間として求めることができる。

$$\mathbf{d}_i^r(u) = \frac{u_{i+n-r}-u}{u_{i+n-r}-u_{i+1}}\mathbf{d}_{i-1}^{r-1}(u) + \frac{u-u_{i-1}}{u_{i+n-r}-u_{i-1}}\mathbf{d}_i^{r-1}(u) \tag{34}$$

なお，$\mathbf{d}_n^n(u)$ は，B スプライン曲線上の点になる。

このド・ブーアのアルゴリズムを適用して，B スプライン曲線のブロッサム $\mathbf{d}[\nu_1, \cdots \nu_n]$ を得ることができる。区間 λ を決めたあと，ド・ブーアのアルゴリズムを適用する際に，1 つのパラメータ値 ν を使う代わりに，異なるパラメータ値 ν_i を使う。

$$\mathbf{d}_i^1[\nu_1] = \frac{\nu_1-u_I}{u_{I+1}-u_I}\mathbf{d}_{i-1} + \frac{u_I-\nu_1}{u_{I+1}-u_I}\mathbf{d}_i \tag{35}$$

$$\mathbf{d}_i^r[\nu_1, \cdots \nu_r] = \frac{u_{i+n-r}-\nu_r}{u_{i+n-r}-u_{i+1}}\mathbf{d}_{i-1}^{r-1}[\nu_1, \cdots \nu_{r-1}] + \frac{\nu_r-u_{i-1}}{u_{i+n-r}-u_{i-1}}\mathbf{d}_i^{r-1}[\nu_1, \cdots \nu_{r-1}] \tag{36}$$

最終的に求まる $\mathbf{d}_n^n[\nu_1, \cdots \nu_n]$ を，単に $\mathbf{d}^{\lambda}[\nu_1, \cdots \nu_n]$ と書く。区間 $\lambda = [u_I, \cdots \nu_{I+1}]$ に定義されるブロッサム $\mathbf{d}^{\lambda}[\nu_1, \cdots \nu_n]$ には，$n+1$ 個の制御点が必要であることが知られている。それらの制御点は，ブロッサム値として得られる。

$$\mathbf{d}_{I-n+1+k} = \mathbf{d}^{\lambda}[u_{I-n+1+k}, \cdots u_{I+k}];\ k = 0, \cdots n \tag{37}$$

（式37）の $\mathbf{d}^{\lambda}$ の括弧内は，u_I か u_{I+1} を含む n 個のノット列になっている。λ に対応するベジエ点 $\mathbf{b}_i$ を求めるのであれば，以下のブロッサムの評価をすればよい。

$$\mathbf{b}_i = \mathbf{d}^{\lambda}[u_I^{\langle n-1\rangle}, u_{I+1}^{\langle i\rangle}];\ i = 0, \cdots n \tag{38}$$

こうして，Bスプライン曲線の制御点から，ベジエ曲線の制御点が求められるため，Bスプライン曲線からベジエ曲線への変換ができる。

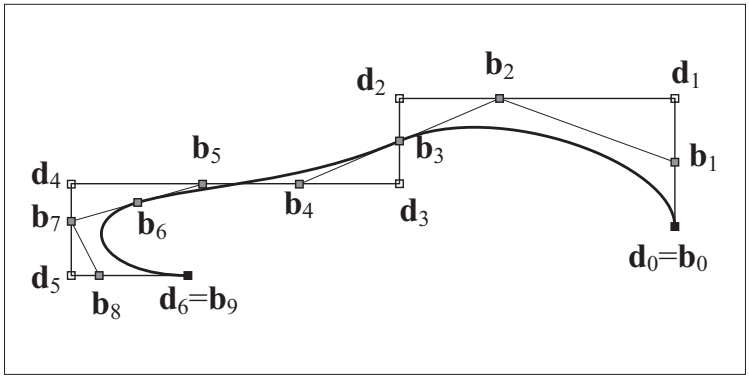

図7　3次のBスプライン曲線を3次のベジエ曲線に変換

2.4. 有理ベジエ曲線とNURBS（非一様有理Bスプライン）曲線

円錐曲線を正確に表現するためにも，有理の多項式曲線が必要とされる。そのため，実際に内部の形式は，すべて有理多項式曲線の形で情報を保持すべきだと考えられる。ベジエとBスプラインの両方の曲線において，有理曲線を定義してゆく。

有理ベジエ曲線はバーンスタイン多項式を用いて，次のように定義される。以下の式において，w_i は，重みと呼ばれる。

$$\mathbf{b}(t) = \frac{\sum_{i=0}^{n} w_i \mathbf{b}_i B_i^n(t)}{\sum_{i=0}^{n} w_i B_i^n(t)} \tag{39}$$

1つの重み w_i が増やされると，曲線は制御点 $\mathbf{b}_i$ に向かって引き寄せられる。有理ベジエ曲線は，次の3つの方法を用いて，設計できる。

・制御点 $\mathbf{b}_i$ を移動する方法
・重み w_i を変更する方法
・補助点（weight point）$\mathbf{q}_i$ を利用する方法

補助点は，次のように計算することができる。

$$\mathbf{q}_i = \frac{(w_i \mathbf{b}_i + w_{i+1} \mathbf{b}_{i+1})}{w_i + w_{i+1}} \tag{40}$$

上記の式は，重みから補助点を計算するものであるが，この式を展開して逆に補助点から，重みを計算することができる。共線上の3点の比 $ratio(\mathbf{a}, \mathbf{b}, \mathbf{c})$ は，$\mathbf{b} = (1-\alpha)\mathbf{a} + \alpha\mathbf{c}$ と仮定すると次のように定義できる。

$$ratio(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \frac{\alpha}{1-\alpha} \tag{41}$$

この比を用いて，重みの比率は次のように表される。

$$ratio(\mathbf{b}_i, \mathbf{q}_i, \mathbf{b}_{i+1}) = \frac{w_{i+1}}{w_i} \tag{42}$$

図 8 においては，菱形の点が補助点を示している。

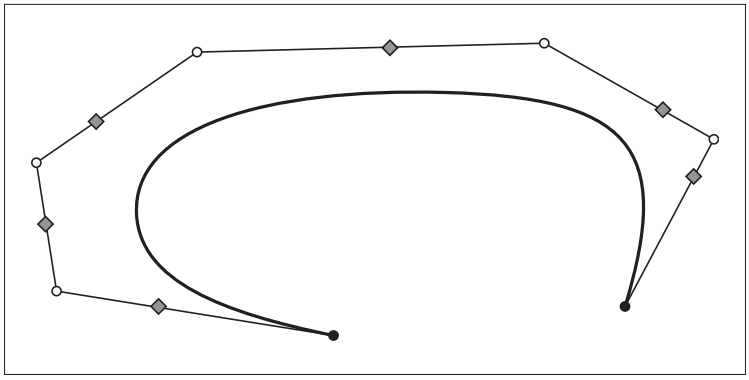

図 8　有理ベジエ曲線の補助点の例

有理Ｂスプライン曲線はＢスプライン関数を用いて次のように定義される。有理ベジエ曲線と同様に，ω_i は，重みである。

$$\mathbf{s}(u) = \frac{\sum_{i=0}^{p} \omega_i \mathbf{d}_i N_i^n(u)}{\sum_{i=0}^{p} \omega_i N_i^n(u)} \tag{43}$$

NURBS 曲線は，ノット列の間隔が一定ではない有理Ｂスプライン曲線のことであり，このノット列の間隔の比率により，曲線のセグメントの区間が決まってくる。それ以外は，(式43) で定義される有理Ｂスプライン曲線と同様である。有理曲線の間の変換については，前節で述べた非有理の曲線の変換と同等である。

2.5. 曲線の次数増加と減少への対応

ベジエ曲線の場合，n 次のベジエ曲線が，$n+1$ 次のベジエ曲線で表現できる。バーンスタイン多項式については，次の 3 つの式で表現される。

$$(1-t)B_i^n(t) = \frac{n+1-i}{n+1} B_i^{n+1}(t) \tag{44}$$

$$tB_i^n(t) = \frac{i+1}{n+1} B_{i+1}^{n+1}(t) \tag{45}$$

$$B_i^n(t) = \frac{n+1-i}{n+1}B_i^{n+1}(t) + \frac{i+1}{n+1}B_{i+1}^{n+1}(t) \tag{46}$$

（式46）から，n 次の有理ベジエ曲線の制御点から $n+1$ 次のベジエ曲線の制御点 $\mathbf{b}_i^{n+1}$ と重み w_i^{n+1} を求めることができる。

$$\mathbf{b}_i^{n+1} = \frac{iw_{i-1}\mathbf{b}_{i-1}^n + (n+1-i)w_i\mathbf{b}_i^n}{iw_{i-1} + (n+1-i)w_i};\ i = 0, \cdots n+1 \tag{47}$$

$$w_i^{n+1} = iw_{i-1}^n + (n+1-i)w_i^n \tag{48}$$

このように，次数を 1 つ上げて曲線を表現する方法は，有理ベジエ曲線の次数上げ（degree elevation）と呼ばれるが，図 9 は有理 3 次ベジエ曲線を有理 4 次ベジエ曲線で表現した例になっている。

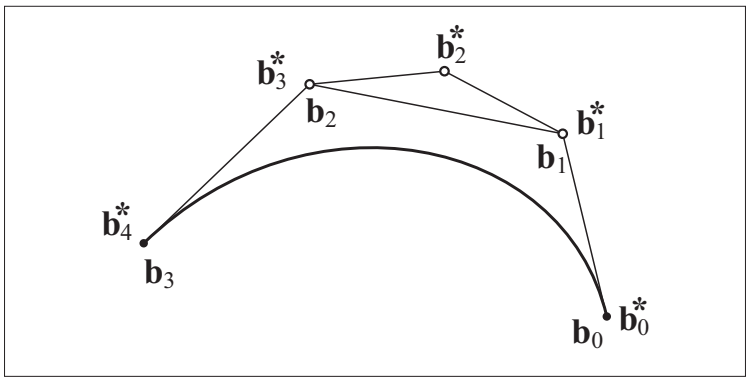

図 9　有理ベジエ曲線の次数上げの例

有理ベジエ曲線では，次数を下げることはできない。次数下げ（degree reduction）は，近似的に行なう[21][22]しかない。

有理Ｂスプライン曲線の場合も，Ｂスプライン多項式の次数上げが次のように記述できる。

$$N_i^n(t) = \frac{1}{n+1}\sum_{j=i-1}^{n+1} N_{i+1}^{n+1}(u; u_j) \tag{49}$$

（式49）において，$N_{i+1}^{n+1}(u; u_j)$ は，元の n 次のＢスプライン曲線のノット列に対して定義されるが，ただし，このノット列ではノット u_j だけは 1 つだけ多重度が増やされている。NURBS 曲線でも，次数を下げることは近似的になるが，様々なアルゴリズムが提案されている[23][24]。

---

[21] Farin, NURBS, 邦訳　pp.123-124.

[22] Farin, CAGD, pp.85-86.

3．描画のためのユーザインタフェースとその実現

描画においては，各曲線のための描画方法を選択する方式を採用することにした。加えて，後から加工変形できるために，別パネルにおいて複合曲線の各セグメント曲線の情報を表示することや，各制御点（制御多角形の頂点）の情報を表示することも行なった。これらは，solidThinking などの3DCG 用に設けられているものを参考にした。加工変形するための，ピボット（動作の中心となる点）設定は，Blender の3D カーソルや重心点などを選ぶことができるようになっている機能を参考にした。また，Adobe Illustrator では各制御点を編集するためのツール（ダイレクト選択ツール）があるが，制御点を編集するモードを別途に追加した。この方式は，3DCG モデリング用ソフトウェアで一般的になっている。以下にそれぞれの操作の詳細について説明してゆく。

3.1．描画

いくつかの描画ツールを用意することにした。各曲線を描く際には，補助パレットにおいて，曲線の次数，および各制御点の重みを設定できるようにしている。以下にそれぞれの描画ツールによる描画方法の概要と実行例の図を挙げてゆく。なお，制御点を置くときに，Shift キーとの組み合わせで，何らかの制約（整列や連続性の設定）が掛かるようにしている。また，これら描画方法で描画された曲線は，有理ベジエ曲線か NURBS 曲線のいずれかとして編集できるように情報が格納されてゆく。

エルミート曲線風の描画：Adobe Illustrator が用いているエルミート曲線風の描画方式で，ベジエ曲線から構成される複合曲線の C^2 連続性を保証するために，描画時は端点に対しての前後の 2 つの制御点のベクトルを共線にし，互いに逆ベクトルにさせるようにした。このベクトルは，描いていくときに端点を描き，そこからドラッグによって接線ベクトルを表示するものである。基本的には 3 次のベジエ曲線を描くために用いられる。これらのベクトルは，制御点編集モードで Adobe Illustrator のように，C^1 連続（ベクトルは共線で反対方向を向いているが大きさは異なる）あるいは C^0 連続（ベクトルは始点だけ同じ端点になっている：Adobe Illustrator ではコーナーポイントと呼んでいる）にさせることも可能である。

ベジエ制御点による描画：これは，3 次であればベジエ曲線の制御点として 4 点（うち 2 つは端点）を指定するものである。次数は，基本的に 3 次に設定されているが，2 次以上の値であれば設定可能になっている。n 次の有理ベジエ曲線の場合，$n+1$ 個の制御点から構成される。この描画方法では，連続描画で複合曲線にする場合でも C^0 接続以上にはならないが，1 つの曲線セグメントを描いた後，次の制御点をクリックするときに Shift キーを併用することにより C^1 接続の形で描けるようにした。

スプライン補間による描画：これは，Autodesk Maya において採用されているものと

---

(23) Les Piegl and Wayne Tiller, Algorithm for degree reduction of B-spline curves, Computer-Aided Design, Volume 27, Issue 2, February 1995, Elsevier, pp.101-110.

(24) Jun-Hai Yonga, Shi-Min Hua, Jia-Guang Suna, and Xing-Yu Tan, Degree reduction of B-spline curves, Computer Aided Geometric Design, Volume 18, Issue 2, March 2001, Elsevier, pp.117-127.

同等で，スプライン補間を用いて，頂点が追加されるたびに，スプライン曲線の形を変化させるものである。2点の頂点を描いた場合は直線になり，3点以上描いて初めて曲線になる。これも次数を設定することができる。

Ｂスプライン制御点による描画：多くの3DCGソフトウェアと同じ方式のマウスのクリックで制御点を置いていく形になる。ノット列も描画に合わせて作られる。補助パネルでノット列の値を変更することができ，この値を整数値から実数値に変えることによって，非一様曲線にすることができる。ただし，値は最初のノットから昇順（1つ前のノットの値より等しいか大きい値）になっていなければならない。また，開いたもの（端点があるもの）と，閉じたもの（楕円のような閉包した曲線）が描画できるようにしてある。曲線を端点に合わせる場合は，自動的にノット列がn個重複して作られる。図10は左から3次のエルミート曲線風の描画，4次のベジエ曲線，同じく4次のＢスプラインの描画の例になっている。

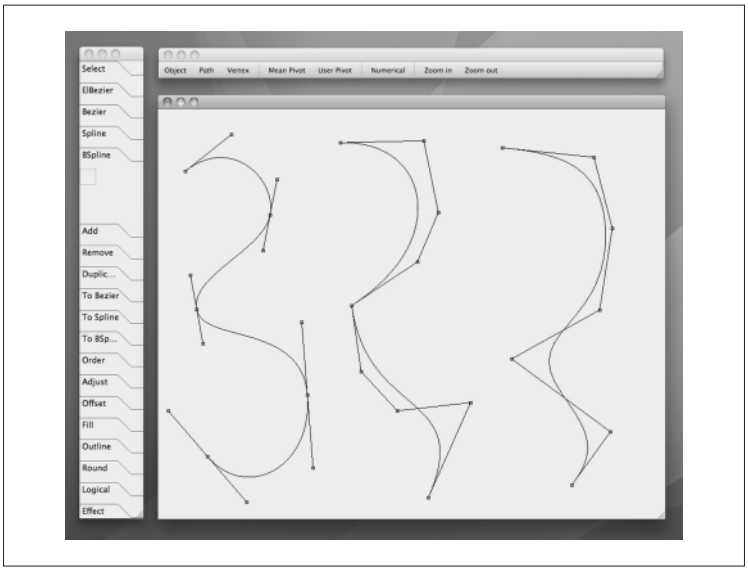

図10　各描画方法による描画例

3.2. 曲線（パス）の加工

曲線の加工は，オブジェクトとしての曲線にはアフィン変換，すなわち，移動・回転・拡大／縮小・鏡像・シアーなどの変形が可能である。これは，他のドローイングソフトに共通のものとなっている。ここでは，それらの一般の加工方法の他に，有理曲線ならでは加工方法について列挙してゆく。

次数の変更：内部的には次数の変更は，次数上げしか対処していない。次数上げの場合は，指定された次数に対応したその曲線を再現するための必要な制御点が増やされる。次数下げをした場合，曲線の形が変わってしまうことは避けがたい。ただし，曲線の滑らかさを上げる，すなわち，ある程度の詳細な部分を割愛して，制御点の個数を減らしていく

操作は，多くのドローイングソフトウェアで行なわれているため，今後はそれに準じた形で，次数下げを近似で行なう方法を模索していきたい。

拡張・接続・統合：拡張とは，既存の曲線から制御点を追加して更に複合曲線を伸ばすこと，接続とは，既存の複数の曲線の間の端点間を指定して結合を行なうこと，そして統合とは結合において既存の複数の曲線の表現方法や次数が異なる場合に，これらを表現方法や次数を揃えた形で統一する操作になる。よって，後に述べる曲線の表現形式の変換の機構も統合には組み込まれている。

タンジェントツールによる変形：タンジェント（tangent）ツールと呼ばれているものは，法線と接線による変更曲線の任意の場所での変形であり，Autodesk Maya および solidThinking などにおいて採用されている。曲線の任意の場所で接線ベクトルと法線ベクトルの修正ができる。見栄えの良いツールではあり，曲線上の任意の場所で直線などに添わせるときに有効ではあるが，今回の試作では見送った。

曲線の表現の変換：2.3．節で記述したもので，ベジエ曲線で描画していたものをＢ－スプライン曲線に変換する，あるいはその逆方向の変換である。

有理ベジエ曲線の補助点による重みの変更：2.4．節に記述したもので，補助点を表示し，それを制御多角形の辺上を動かすことによって，重みの調整をするものである。

プリミティブ図形からの有理曲線化：楕円や円などの円錐曲線を用いたプリミティブ図形を描画できるツールはどのドローイングソフトウェアでも用意されているが，それを有理ベジエ曲線あるいは NURBS 曲線に変換するものである。

その他，制御点編集モードでは，選択された制御点だけをアフィン変換する加工機能や，選択された制御点だけを軸に揃えて整列する，あるいは均等に分散して配置する機能などが必要とされている。

4．試作とおわりに

プロトタイプは，Python[25] スクリプティング言語を用いて，C/C＋＋用のグラフィックユーザインタフェースライブラリである Qt4[26] を Python 用に移植した PyQt4[27] ライブラリを用いて試作を行なった。PyQt4ライブラリもそうであるが，２次元グラフィックス描画ライブラリのほとんどは，３次のベジエ曲線による描画関数を保持している。しかしながら，今回の実装においては，独自の有理ベジエ曲線の描画関数と NURBS 曲線の描画関数を用意した。これは，３次以上の高次でかつ有理の多項式曲線の描画に対応するためである。内部では，有理ベジエ曲線と，有理Ｂスプライン曲線の２つの形式で曲線情報を保持している。異なる表現形式の曲線が結合されたとき，C^0 接続の場合には，それぞれの情報のままで保持される。C^1 あるいは C^2 接続の場合は，プロンプトを出し，どちらの形式に統一するかユーザに尋ねることにした。今後は，iOS に移植することも考えて，Cocoa[28] ライブラリで再実装を予定している。

---

[25] Python Software Foundation, Python Programming Language, http://www.python.org

[26] Nokia, Qt-Cross-platform application and UI framework, http://qt.nokia.com/

[27] Riverbank computing, PyQt4 Manual, http://www.riverbankcomputing.co.uk/

[28] Cocoa-Mac OS X Technology Overview, Apple, https://developer.apple.com/technologies/mac/cocoa.html

3 次のベジエ曲線は，2 次元のドローイングソフトウェアの代表である Adobe Illustrator や Canvas などで用いられている他，通常の 2 次元グラフィックユーザインタフェースのライブラリやプリンタの印刷用の言語である PostScript などでも用いられている。しかし，高次の有理ベジエ曲線は，一般的には普及していない。この試作物などを配布することで，高次の有理ベジエ曲線を利用した表現の普及を促進したい。そこからは，新たな曲線表現が現れてくるだろう。また，NURBS 曲線は，3DCG モデリングソフトウェアで用いられているが，曲線表現自体への拘りというものではなく，CAD 上の技術であるからという位置づけになっている。3DCG モデリングソフトウェアでも，Strata 3D や Shade，Carrara3D といった安価なソフトウェアでは，Adobe Illustrator と同等のベジエ曲線の描画方法を曲線描画に主として用いている。Autodesk Maya ではベジエ曲線を描くツールが NURBS 曲線描くツールとは別に用意されている。Blender，solidThinking，Rhinoceros 3D などを含め CAD 系のソフトウェアでは，NURBS 曲線を描くことでモデリングを行なうが，それに対してベジエ曲線としての扱いをすることはできない。高次の有理ベジエ曲線と NURBS 曲線がお互いに変換可能なことは，2.3．節に示したように数式的には可能であるが，それらの曲線を混在させて様々な表現を行なっていくことは，今後もっと行なわれて然るべきであろう。まずは 2 次元から試作を行なったが，このソフトウェアを実用のものとして完結させた後は，3 次元モデリングについても同じような試みを続けていきたい。また，iPad などのタブレット機で実装する際には，タッチパネルを複数の指を用いて描画表現することが可能になっている。このような表現において，両方の曲線表現を用いて，2 次元のドローイングをどのように行なっていくのかについても継続的な研究として行なっていきたい。

本稿は平成22年度千葉商科大学学術研究助成金による研究の成果である。

〔抄　録〕

2 次元描画においては，Adobe 社の Illustrator を始めとして，ベジエ（Bézier）曲線が使われている。しかしながら，3 次元描画においては，特定の 3 次元モデリングソフトウェアでは，ベジエ曲線の形で曲線描画ができるが，これらは必ずしも正確なモデリングができるとは言い難い。正確なモデリングを行なう CAD 系のソフトウェアでは，非一様有理Ｂスプライン（NURBS）曲線が使われていることが多い。数学的には，既にベジエ曲線とＢスプライン曲線を相互に変換する方法が知られている。また有理曲線についても，有理ベジエ曲面と有理スプライン曲面を変換することも可能となっている。歴史的な要因が強いと思われる 2 次元でのベジエ指向の偏向にバランスを取るべく，2 次元描画ソフトウェアにおいて，ベジエ曲線とＢスプライン曲線を併用して用いることができるソフトウェアを試作し，3 次元描画ソフトウェアで行なわれている加工変形の機能を，2 次元描画ソフトウェアにも組み込んでみる。本論文は，平成22年度千葉商科大学学術研究助成金による研究の成果である。