５．　Bスプライン(B-Spline)

５．１ B スプライン関数

（定義）

ノットベクトルと呼ばれる(n+k)個の非減少な実数列 $\boldsymbol{t} = (t_j)_{j=0}^{n+k-1}$ が与えられたとき、ノットベクトル $\boldsymbol{t}$ に対するオーダ k の i 番目の B スプラインは次式で定義される：

（式2-1）　$B_{i,k,t}(x) = (t_{i+k} - t_i)[t_i,...,t_{i+k}](\cdot - x)_+^{k-1}$　　for i=0,…,n-1

ここで、$[t_i,...,t_{i+k}](\cdot - x)_+^{k-1}$ は関数 $(\cdot - x)_+^{k-1}$ の差分商(divided difference)。関数 $(\cdot - x)_+^{k-1}$ は切断冪関数(truncated power function)。

切断冪関数 $(\cdot - x)_+^{k-1}$ は次のように定義される。ここでは汎関数である差分商のパラメータとしているのでその変数は特に重要ではなく、プレースホルダー　$(\cdot)$　で表現している― x はBスプラインの変数であるが、切断冪関数の変数ではない。次の説明ではこれを表現するのに、変数 χ を用いている。

（式2-2）　$(\cdot - x)_+ = (\chi - x)_+ = \begin{cases} \chi - x & \chi \geq x \\ 0 & \chi < x \end{cases}$

（式2-3）　$(\cdot - x)_+^{k-1} = ((\cdot - x)_+)^{k-1}$

以上を用いて差分商 $[t_i,...,t_{i+k}](\cdot - x)_+^{k-1}$ は次のように定義される。

（式2-4）　$[t_i,...,t_{i+k}](\cdot - x)_+^{k-1} = \dfrac{[t_{i+1},...,t_{i+k}](\cdot - x)^{k-1} - [t_i,...,t_{i+k-1}](\cdot - x)^{k-1}}{t_{i+k} - t_i}$

$$\vdots$$

（式2-5）　$[t_i, t_{i+1}](\cdot - x)_+^{k-1} = \dfrac{(t_{i+1} - x)_+^{k-1} - (t_i - x)_+^{k-1}}{t_{i+1} - t_i}$

$B_{i,k,t}(x)$ はノットベクトルを明示する必要がない場合は $B_{i,k}(x)$ と記述され、さらにオーダも明示する必要がない場合は $B_i(x)$ と簡略化して記述される。

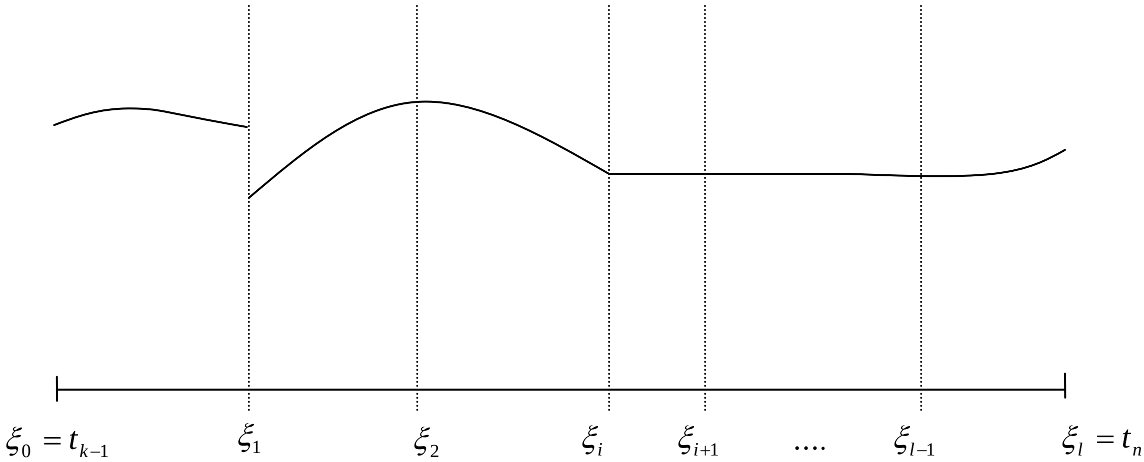
５．２ B スプラインの性質１

$0 \leq B_i(x) \leq 1$ である。

あるパラメータ値のBスプライン関数の総和は１である。すなわち、パラメータ値 x（$t_{k-1} \leq x < t_n$）が与えられたとき、$\displaystyle\sum_{i=0}^{n-1} B_i(x) = 1$

B スプライン列$\{B_i(x)\}_0^{n-1}$は$t_\mu \le x < t_{\mu+1}$である x が与えられれば　$i < \mu - k + 1$、または$\mu < i$である$B_i(x)$はすべて 0（ゼロ）であり、k 個の B スプライン関数$\{B_i(x)\}_{i=\mu-k+1}^{\mu}$だけが非ゼロの可能性がある。また、$\sum_{i=\mu-k+1}^{\mu} B_i(x) = 1$である。

５．３ Curry, Shoenberg の定理(PP 表現の B スプラインによる表現)



単調増加な実数列　$\xi = (\xi_i)_0^l$（分割点-break points と呼ばれる）とξ内部の各分割点ξ_i $1 \le i \le l-1$における連続性を表現する整数値$\nu = (\nu_i)_1^{l-1}$、　$1 \le \nu_i \le k$（ξ_iにおける連続性は$C^{(k-\nu_i-1)}$）for $1 \le i \le l-1$が与えられ、$n = k + \sum_{i=1}^{l-1} \nu_i$とする。また、$\boldsymbol{t}$= $(t_j)_{j=0}^{n+k-1}$を次の条件を満たす非減少な実数列とする

$t_0 \le t_1 \le .... \le t_{k-1} \le \xi_0$、かつ、$\xi_l \le t_n \le .... \le t_{n+k-1}$

i=1,...,$l-1$に対して、ξ_iはベクトル $\boldsymbol{t}$ 内でν_i回現れる

このとき、オーダk（または次数$k-1$）で、ノットベクトル $\boldsymbol{t}$ の B スプライン列$(B_i)_{i=0}^{n-1}$（B_0,...,B_{n-1}）は実数区間$[t_{k-1}, t_n]$をパラメータ区間とする関数空間$P_{k,\xi,\nu}$の基底(basis)である。　ここで、$P_{k,\xi,\nu}$はブレークポイント$\xi = (\xi_i)_0^l$上の各点における連続性の定義を伴った、オーダkの

区分多項式関数空間。$P_{k,\xi,\nu}$ は、オーダ k 、Ｂ表現次元ｎ、ノットベクトル $\boldsymbol{t}$ から表現される区分多項式表現全体が構成する（関数空間としての）線形空間である。
別の言い方をすれば、

オーダ（多項式の次数を定義する）

分割点のパラメータ値

各分割点における連続性

の三つが定義された区分的多項式表現の関数は、すべてＢスプライン列 $(B_i)_{i=0}^{n-1}$ の一次結合により一意的に表現可能である。

５．４ Ｂ表現の定義

多項式のオーダを表現するｋおよび（関数空間としての）線形空間の次元数を表現する
$n = k + \sum_{i=1}^{l-1} v_i$ （Ｂ表現次元と呼ばれる）

ノットベクトル t = $(t_j)_{j=0}^{n+k-1}$

Ｂスプライン列 $(B_i)_0^{n-1}$ に関する、関数 f の係数ベクトル $\alpha = (\alpha_i)_0^{n-1}$ （制御点列（Control Points）と呼ばれる）

関数 $f \in P_{k,\xi,\nu}$ に対するＢ表現は上の各データから次のように定義される：

（式2-6）　$$f(x) = \sum_{i=0}^{n-1} \alpha_i B_i(x)$$

特に　$k-1 \le \mu < n$ であるμに対して $t_\mu \le x < t_{\mu+1}$ であれば、

（式2-7）　$$f(x) = \sum_{i=0}^{n-1} \alpha_i B_i(x) = \sum_{i=\mu-k+1}^{\mu} \alpha_i B_i(x)$$

である。すなわち、Ｂスプライン列 $(B_i)_0^{n-1}$ は $t_\mu \le x < t_{\mu+1}$ であれば $i < \mu - k + 1$ 、または $\mu < i$ である $B_i(x)$ はすべて０（ゼロ）となる。ｋ個の Ｂスプライン $(B_i)_{i=\mu-k+1}^{\mu}$ だけが非ゼロの可能性がある。

ノットベクトル t ＝ $(t_j)_{j=0}^{n+k-1}$ にはＢ表現次元 n、オーダ k の情報が含まれるとして、Ｂ表現は

ノットベクトルと制御点列の対（t、α）により表現されると考えてよい。

５．５ B スプラインの性質２

Ｂスプライン関数は強凸包（strict convex hull）である。x が $t_\mu \le x < t_{\mu+1}$ を満たすようなパラメータ値であるとき、$f(x) = \sum_{i=\mu-k+1}^{\mu} \alpha_i B_i(x)$ であり、i=μ-(k-1), μ-(k-2),..., μ の k 個の制御点 α_i だけの一次結合で表わされる。

Ｂ表現は変動減少(Variation Diminishing)特性を持つ。Ｂ表現は制御点列を直線で結んだ多角形（定義多角形）の変動よりも変動が少ない。

５．６ B スプライン関数の評価－１

オーダ k のスプライン列 $(B_{i,k})_{i=0}^{n-1}$ は、$t_\mu \le x < t_{\mu+1}$ であるパラメータ値 x が与えられた時、

すなわち、

パラメータ値 x が与えられ

x がノットベクトルの $t_\mu \le x < t_{\mu+1}$ を満たすような、どこに位置するかを求め

そのノットベクトルの位置をμとする時

k 個の非ゼロである（非ゼロの可能性のある）B スプライン列 $\{B_{i,k}(x)\}_{i=\mu-k+1}^{\mu}$ は、次のように

帰納的に求めることができる。

（式2-8） $$B_{\mu,1}(x) = \begin{cases} 1 & for \quad t_\mu \le x < t_{\mu+1} \\ 0 & for \quad x < t_\mu \quad or \quad t_{\mu+1} \le x \end{cases}$$

（式2-9） $$B_{i,j+1}(x) = \frac{x - t_i}{t_{i+j} - t_i} B_{i,j}(x) + \frac{t_{i+j+1} - x}{t_{i+j+1} - t_{i+1}} B_{i+1,j}(x)$$

$i = \mu - j, \ldots, \mu$ for j=1,…k-1

（式 2-9）を、 a を添え字として、書き直せば、

（式2-10） $$B_{\mu-j+a,j+1}(x) = \frac{x - t_{\mu-j+a}}{t_{\mu+a} - t_{\mu-j+a}} B_{\mu-j+a,j}(x) + \frac{t_{\mu+1+a} - x}{t_{\mu+1+a} - t_{\mu+1-j+a}} B_{\mu-j+a+1,j}(x)$$

$a = 0,..., j$ for j=1,…k-1

MGKnotVector::eval_coef()では(nderiv=0 として)、非ゼロの可能性のある k 個の係数を領域 coef[.]に

coef[a]= $B_{\mu+1-k+a,k}(x)$ for a=0, …, k-1

として求めている。

５．７ B スプライン関数の評価－２（微分値の評価）

B スプライン関数の微分は$t_\mu \le x < t_{\mu+1}$ であるとき

（式2-11） $$f'(x) = (\sum_{i=0}^{n-1} \alpha_i B_{i,k}(x))' = (\sum_{i=\mu-k+1}^{\mu} \alpha_i B_{i,k}(x))' = \sum_{i=\mu-k+2}^{\mu} (k-1)\frac{\alpha_i - \alpha_{i-1}}{t_{i+k-1} - t_i} B_{i,k-1}(x)$$
$$= \sum_{i=\mu-k+1}^{\mu} \alpha_i (k-1)(\frac{B_{i,k-1}(x)}{t_{i+k-1} - t_i} - \frac{B_{i+1,k-1}(x)}{t_{i+k} - t_{i+1}})$$

すなわち

（式2-12） $$B'_{i,k}(x) = (k-1)(\frac{B_{i,k-1}(x)}{t_{i+k-1} - t_i} - \frac{B_{i+1,k-1}(x)}{t_{i+k} - t_{i+1}}) \quad i = \mu - k + 1,..., \mu$$

一般に微分の回数を nd(nd>=k の時、評価結果はすべて 0 であり、nd<k であるとしている)としたとき、

（式2-13） $$B^{(0)}_{i,k-nd}(x) = B_{i,k-nd}(x), \qquad i = \mu - (k - nd) + 1,..., \mu$$

（式2-14） $$B^{(1)}_{i,k-nd+1}(x) = (k-nd)(\frac{B^{(0)}_{i,k-nd}(x)}{t_{i+k-nd} - t_i} - \frac{B^{(0)}_{i+1,k-nd}(x)}{t_{i+k-nd+1} - t_{i+1}})$$

$$i = \mu - (k - nd),..., \mu$$

$$\vdots$$

（式2-15） $$B^{(nd)}_{i,k}(x) = (k-1)(\frac{B^{(nd-1)}_{i,k-1}(x)}{t_{i+k-1} - t_i} - \frac{B^{(nd-1)}_{i+1,k-1}(x)}{t_{i+k} - t_{i+1}}) \quad i = \mu - k + 1,..., \mu$$

（式 2-13）（式 2-14）（式 2-15）を、a を添え字として書き直すと、

（式2-16） $$B^{(0)}_{\mu+1-k+nd+a,k-nd}(x) = B_{\mu+1-k+nd+a,k-nd}(x), \quad a = 0,..., k - nd - 1$$

（式2-17） $B^{(1)}_{\mu-k+nd+a,k-nd+1}(x)=(k-nd)(\frac{B^{(0)}_{\mu-k+nd+a,k-nd}(x)}{t_{\mu+a}-t_{\mu-k+nd+a}}-\frac{B^{(0)}_{\mu+1-k+nd+a,k-nd}(x)}{t_{\mu+1+a}-t_{\mu+1-k+nd+a}}),$

$$a=0,\dots,k-nd$$

$$\vdots$$

（式2-18） $B^{(nd)}_{\mu+1-k+a,k}(x)=(k-1)(\frac{B^{(nd-1)}_{\mu+1-k+a,k-1}(x)}{t_{\mu+a}-t_{\mu+1-k+a}}-\frac{B^{(nd-1)}_{\mu+2-k+a,k-1}(x)}{t_{\mu+1+a}-t_{\mu+2-k+a}})$

$$a=0,\dots,k-1$$

MGKnotVector::eval_coef では（式 2-16）のオーダ(k-nd)の B スプライン関数の値を、(k-nd)個領域 coef[.]に求めた後、（式 2-18）により、

coef[a]= $B^{(nd)}_{\mu+1-k+a,k}(x)$ for a=0, …, k-1

として求めている。

プログラムリスト1　Program code MGKnotVector::eval_coef

```
#define MAX_ORDER 20
// Function's return value id is the index of B-coefficients that
// should be multiplied to.
// coef[a] is for (id+a)-th B-Coefficients, 0<= a <=order-1.
// Multiplication with coef should be done like:
//
// data=0.;
// for(int a=0; a<order; a++) data+=coef[a]*B_coef[id+a].
//
//left indicates whether left continuous(left=true), or right continuous
//(left=false) evaluation.
int MGKnotVector::eval_coef(
   double x,       // Parameter value to evaluate
   double* coef,  // coef's to mutiply are returned.
   unsigned nderiv,//order of derivative, =0 means position
   int left          //Left continuous(left=true)
                     //or right continuous(left=false).
)const{
   int k=m_order;
   int mu=locate(x,left);// k-1<= mu <=bdim()-1 is guaranteed.
   int mup1=mu+1;
   int id=mup1-k;
   if(nderiv>=unsigned(k)){
```

```
    for(int a=0; a<k; ++a)
        coef[a]=0.f;
    return id;
}

double deltal_buf[MAX_ORDER], deltar_buf[MAX_ORDER];
double* deltal=deltal_buf; double* deltar=deltar_buf;
if(k>MAX_ORDER){
    deltal=new double[k]; deltar=new double[k];
}

//deltar[j] will be t[mu+j+1]-x
//deltal[j] will be x-t[mu-j]     for j=0, ... ,k-1-nderiv
const MGKnotVector& t=*this;
int kmnd = k-nderiv;
coef[0] = 1.f;//=Bmu,1(x)
int j;
for(j=1; j<kmnd; j++){
    int jm1=j-1;
    int mup1mj = mup1-j;
    deltar[jm1] = t[mu+j]-x;
    deltal[jm1] = x-t[mup1-j];
    double saved = 0.f;
    for(int a=0; a<j; ++a) {
        double term = coef[a]/(t[mup1+a]-t[mup1mj+a]);
        coef[a] = saved+deltar[a]*term;
        saved = deltal[jm1-a]*term;
    }
    coef[j] = saved;
//Here coef[a] contains Bmu-j+a,j+1(x)
//(B-coefficient of (mu-j+a)th and of order j+1) for a=0, ... , j.
}
if(k>MAX_ORDER){
    delete[] deltal; delete[] deltar;
}

//NOW B-SPLINE coef(a) 0<=a<=K-JDERIV(ORDER OF K-JDERIV)-1 ARE OBTAINED
//GET DERIVATIVE PART OF THE COEFFICIENTS.
//Here j == kmnd, coef[a] contains Bmu-k+nderiv+1+a,k-nderiv(x)
//(B-coefficient of (mu-k+nderiv+1+a)th and of order k-nderiv)
//for a=0, ... , k-nderiv-1.
for(;j<k;j++){
    double fj = double(j);
```

```
        int mup1mj = mup1-j;
        double saved = 0.f;
        for(int a=0; a<j; ++a){
            double term=coef[a]*fj/(t[mup1+a]-t[mup1mj+a]);
            coef[a] = saved-term;
            saved =term;
        }
        coef[j] = saved;
    }

    return id;
}
```

５．８ ベージェ曲線(Bezier Spline)