コンピュータグラフィックス基礎

第6回 曲線・曲面の表現
「Ｂスプライン曲線」

三谷 純

学習の目標

- ベジェ曲線の一般化
- Bスプライン曲線を学ぶ
- 曲線の基底関数の概念を理解する
- 制御点を入力することで、Bスプライン曲線を描画するアプリケーションの開発を行えるようになる

前回の課題

```
// ベジェ曲線の描画
glColor3d(0.0, 0.0, 0.0);
glLineWidth(2);
glBegin(GL_LINE_STRIP);
for(unsigned int i = 0; i+3 < g_ControlPoints.size(); i+=3) {
		for(double t = 0; t <= 1; t+= 0.01) {
				Vector2d p = (1-t)*(1-t)*(1-t) * g_ControlPoints[i] +
						(3*t*(1-t)*(1-t)) * g_ControlPoints[i+1] +
						(3*t*t*(1-t)) * g_ControlPoints[i+2] +
						t*t*t * g_ControlPoints[i+3];
				glVertex2d(p.x, p.y);
		}
}
glEnd();
```

前回の課題について

- 単位法線ベクトルの求め方
  1. 接線ベクトルを求める
  2. 接線ベクトルを正規化して単位接線ベクトルとする
  3. 単位接線ベクトルを90度回転させる

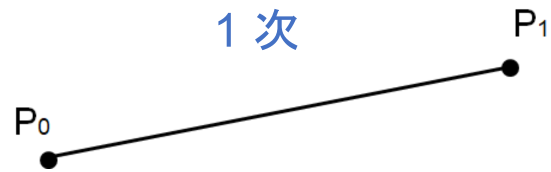
- 接線ベクトルの求め方
  (1) 解析的アプローチ：曲線の式を微分して求める

$$S(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

$$\frac{dS(t)}{dt} = -3(1-t)^2 P_0 + (9t^2 - 12t + 3)P_1 + (-9t^2 + 6t)P_2 + 3t^2 P_3$$

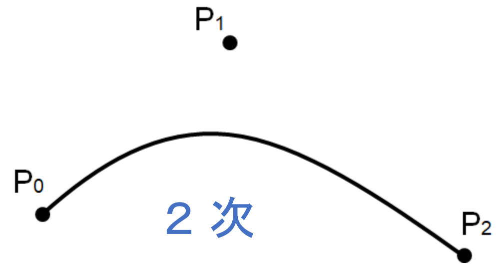
  (2) 数値計算による近似（曲線の式が簡単に微分できない場合）
  - 接線方向＝その点における曲線の進む方向を直線で表したもの
  - 接線方向≒$f(t + \Delta t) - f(t)$

ベジェ曲線の数式表現（復習）



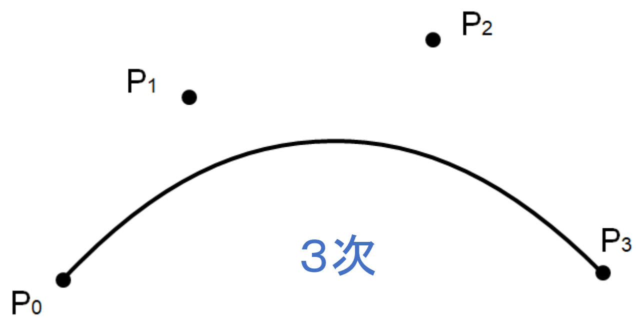
- 1 次

$$\mathbf{P}(t) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1$$



- 2 次

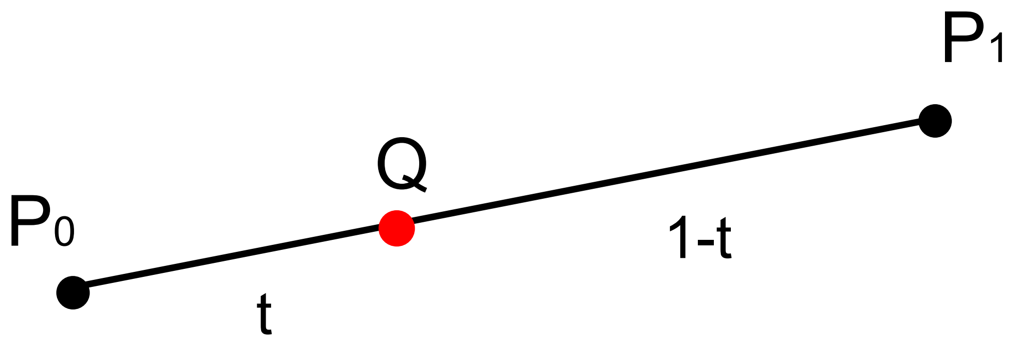
$$\mathbf{P}(t) = (1-t)^2\mathbf{P}_0 + 2t(1-t)\mathbf{P}_1 + t^2\mathbf{P}_2$$



- 3 次

$$\mathbf{P}(t) = (1-t)^3\mathbf{P}_0 + 3t(1-t)^2\mathbf{P}_1 + 3t^2(1-t)\mathbf{P}_2 + t^3\mathbf{P}_3$$

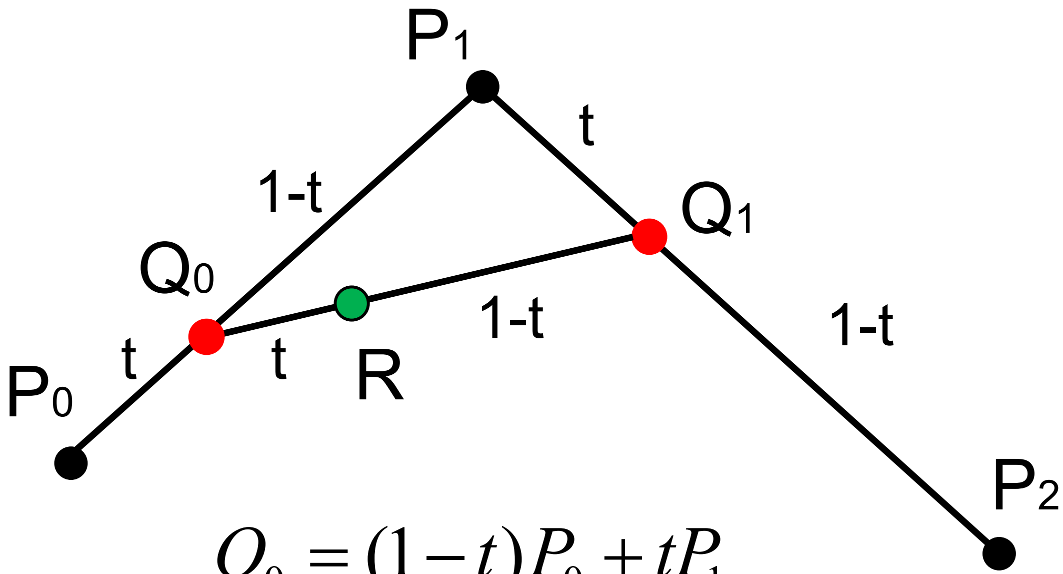
ベジェ曲線の図形的理解（1 次）

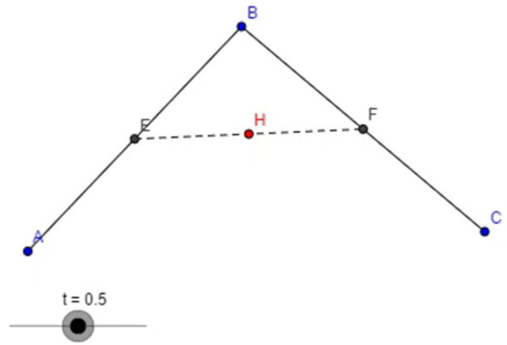


$$\mathbf{Q} = (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \quad (0 \le t \le 1)$$

・t の 1 次式
・2 つの制御点
・直線

ベジェ曲線の図形的理解（2 次）





$$Q_0 = (1-t)P_0 + tP_1$$

$$Q_1 = (1-t)P_1 + tP_2 \qquad (0 \le t \le 1)$$

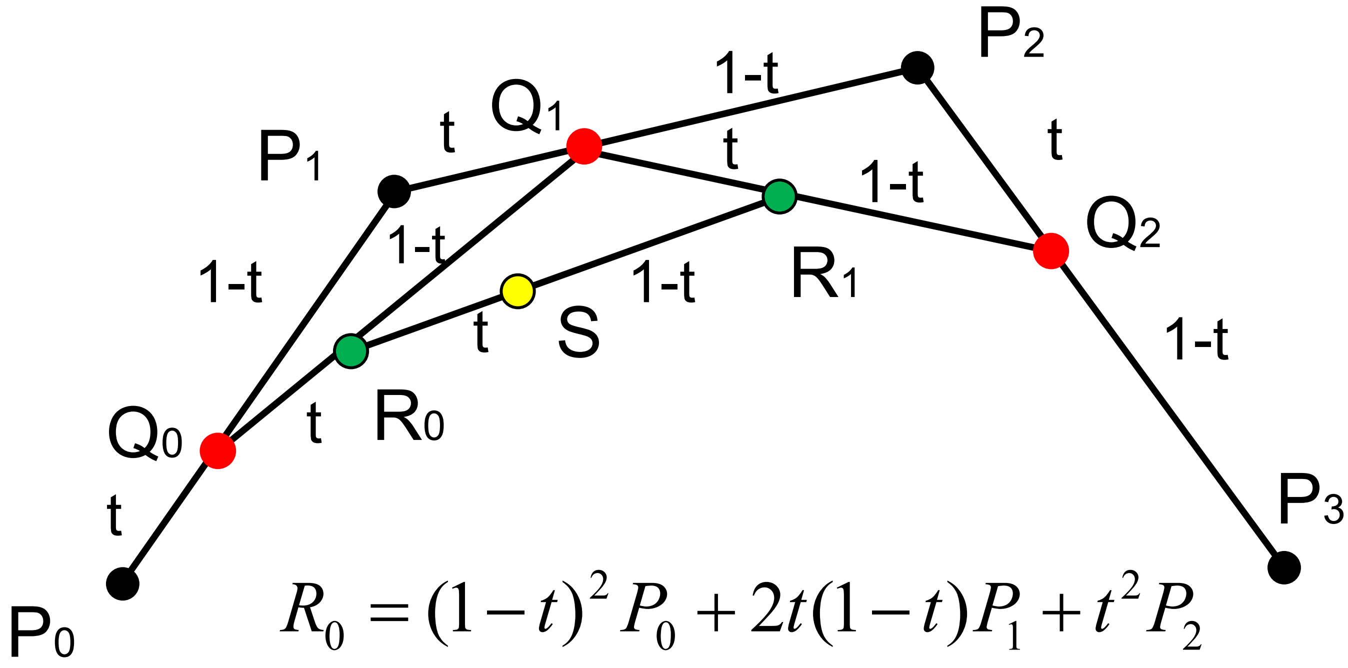
$$R = (1-t)Q_0 + tQ_1$$

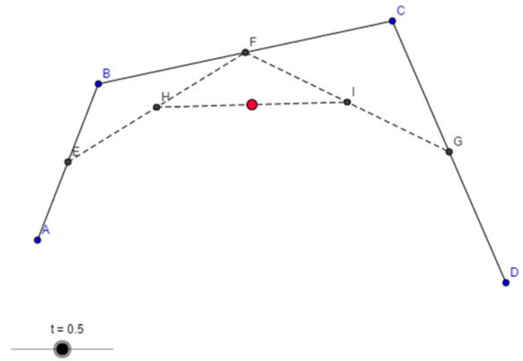
$$\mathbf{R} = (1-t)^2\mathbf{P}_0 + 2t(1-t)\mathbf{P}_1 + t^2\mathbf{P}_2$$

・t の 2 次式
・3 つの制御点
・2 次曲線

（問　$t = 0, 0.5, 1$ のときの位置は？）

ベジェ曲線の図形的理解（3 次）





$$R_0 = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2$$

$$R_1 = (1-t)^2 P_1 + 2t(1-t)P_2 + t^2 P_3$$

$$S = (1-t)R_0 + tR_1$$

$(0 \le t \le 1)$

- t の 3 次式
- 4 つの制御点
- 3 次曲線

$$\mathbf{R} = (1-t)^3\mathbf{P}_0 + 3t(1-t)^2\mathbf{P}_1 + 3t^2(1-t)\mathbf{P}_2 + t^3\mathbf{P}_3$$

ベジェ曲線の係数

３次ベジェ曲線

$$\mathbf{P}(t) = \underline{(1-t)^3}\mathbf{P}_0 + \underline{3t(1-t)^2}\mathbf{P}_1 + \underline{3t^2(1-t)}\mathbf{P}_2 + \underline{t^3}\mathbf{P}_3$$

この係数に見覚えは無いか？

$1 - t = a$ とおいてみると

$$\mathbf{P}(t) = \underline{a^3}\mathbf{P}_0 + \underline{3a^2t}\mathbf{P}_1 + \underline{3at^2}\mathbf{P}_2 + \underline{t^3}\mathbf{P}_3$$

二項係数！

$$(a+t)^3 = a^3 + 3a^2t + 3at^2 + t^3$$

$$\mathbf{P}(t) = \sum_{i=0}^{3} \underline{{}_3\mathrm{C}_i}\, t^i (1-t)^{3-i}\mathbf{P}_i$$

$${}_nC_i = \frac{n!}{i!\,(n-i)!}$$

n 次ベジェ曲線（ベジェ曲線の一般化）

$$\mathbf{P}(t) = \sum_{i=0}^{n} {}_{n}\mathrm{C}_{i}\, t^{i}(1-t)^{n-i}\mathbf{P}_{i}$$

$${}_{n}C_{i} = \frac{n!}{i!\,(n-i)!}$$

2項係数

$\begin{pmatrix} n \\ i \end{pmatrix}$ と表記することもある

さらなる一般化

$$\mathbf{P}(t) = \sum_{i=0}^{n} B_{i}^{n}\mathbf{P}_{i}$$

$$B_{i}^{n} = {}_{n}\mathrm{C}_{i}\, t^{i}(1-t)^{n-i}$$

ある**比率**で各制御点の座標を混ぜ合わせる！

混合比（和は 1 になる）

混合比を関数で表したものを **「基底関数」** とよぶ

基底関数の理解

曲線上の点の位置は、パラメータ $\boldsymbol{t}$ と制御点の位置によって定義される。

曲線上の点の位置は、制御点の座標を混ぜ合わせて作る。

$$\mathbf{P}(t) = \alpha(t)\mathbf{P}_0 + \beta(t)\mathbf{P}_1 + \gamma(\mathrm{t})\mathbf{P}_2 \cdot\cdot$$

この混ぜ合わせ方の係数関数α、β、γ・・・を定義したものが基底関数。

ベジェ曲線の基底関数

- バーンスタイン基底関数

$$B_i^n = {}_n\mathrm{C}_i\, t^i (1-t)^{n-i}$$

n は次数を i は制御点の番号を表す

3 次の場合

$$B_0^3 = (1-t)^3$$

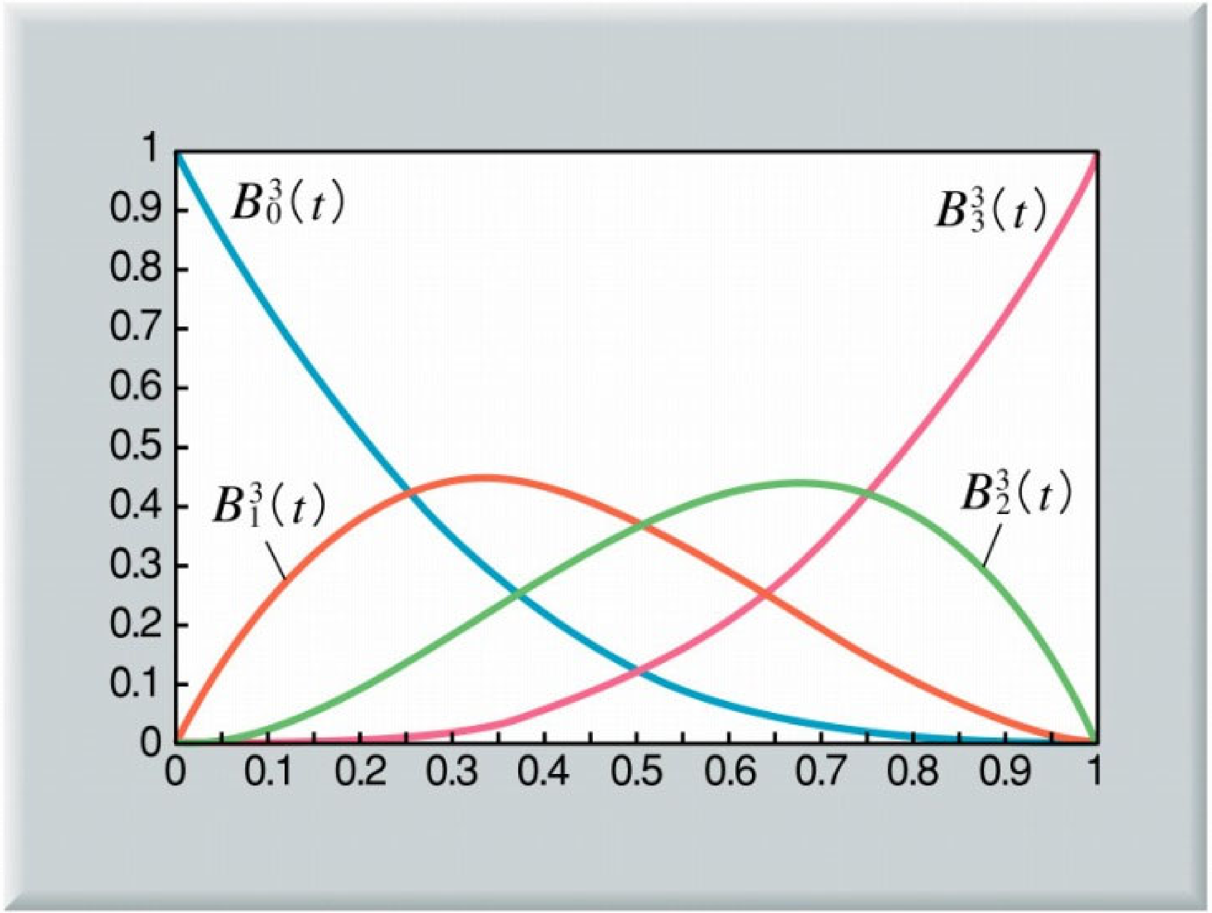
$$B_1^3 = 3t(1-t)^2$$

$$B_2^3 = 3t^2(1-t)$$

$$B_3^3 = t^3$$

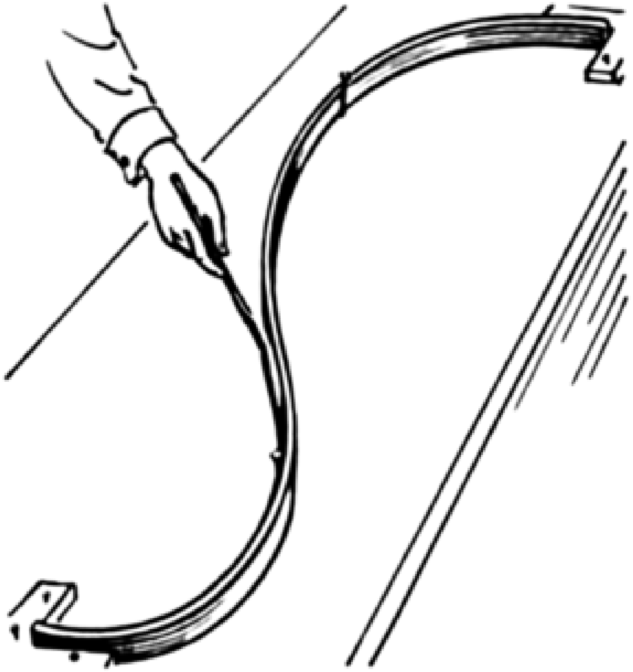
3次のベジェ曲線において
制御点$\mathbf{P}_i$の重み付けが
どのように変化するかを表す

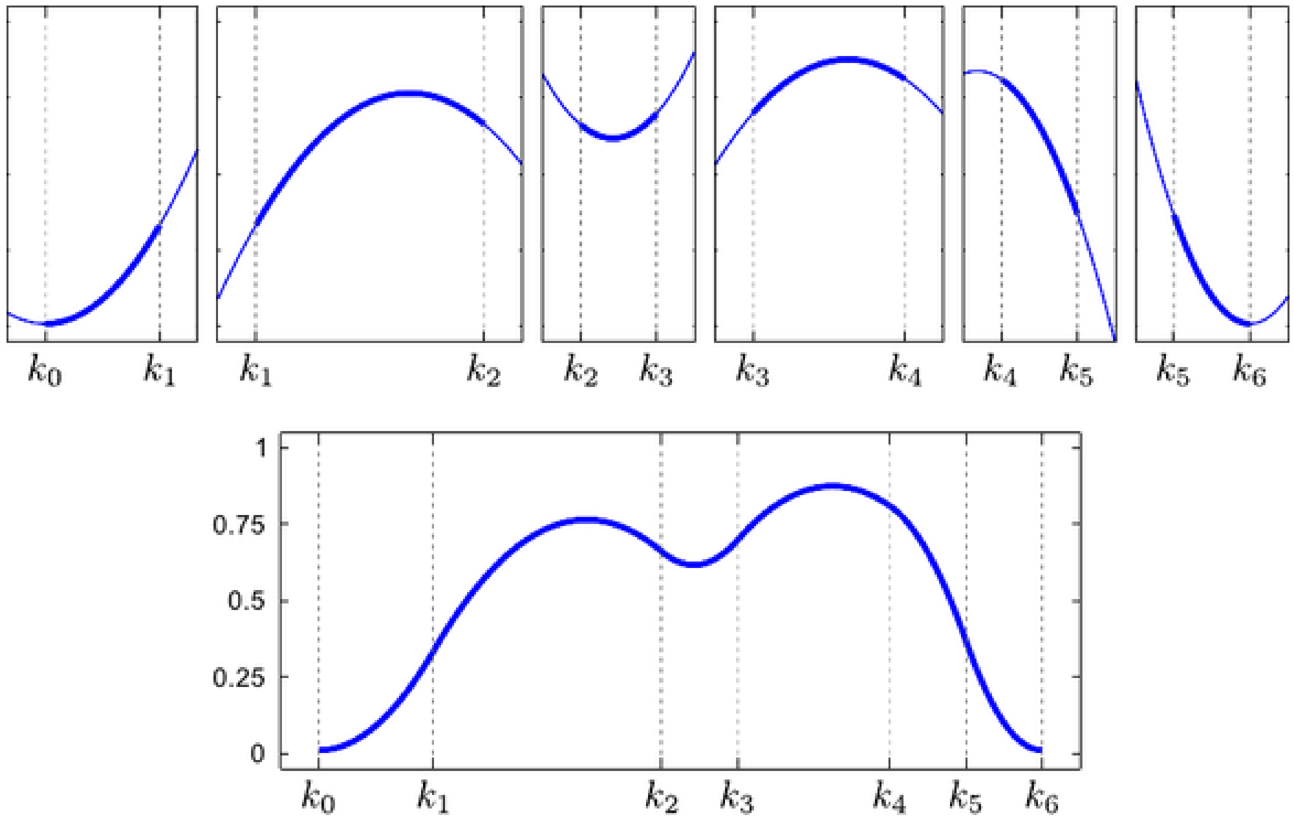
■図3.24——3次バーンスタイン基底関数のグラフ



Ｂスプライン曲線

スプライン曲線のイメージ

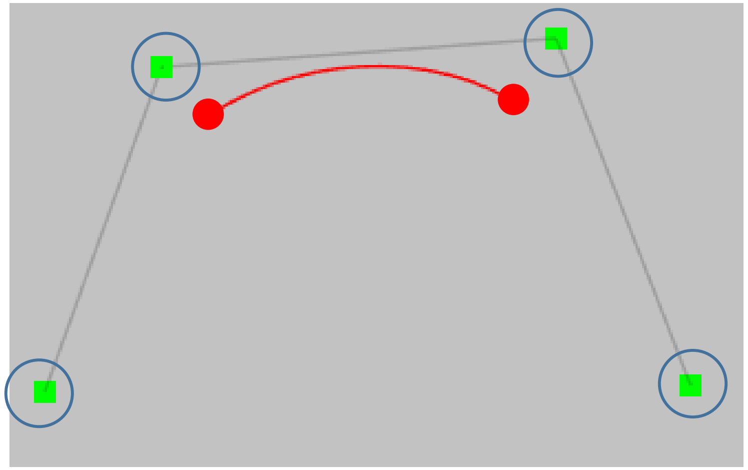


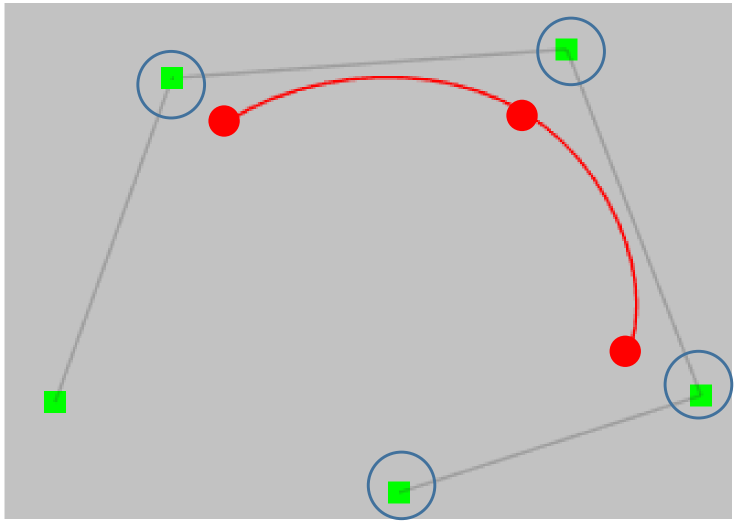


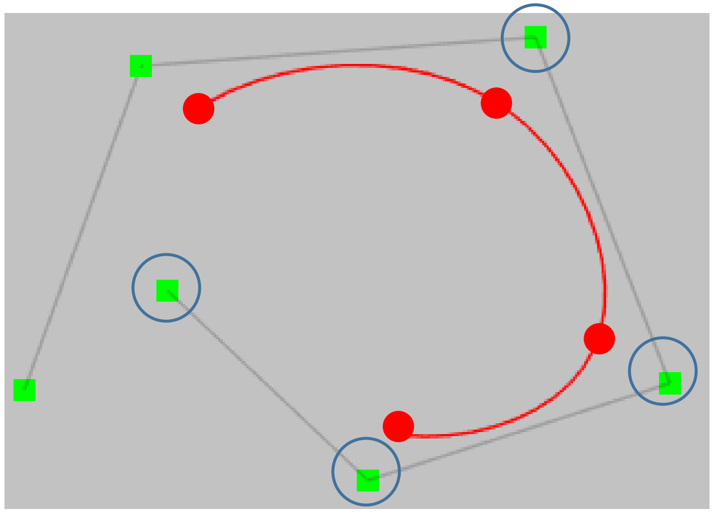
複数の曲線セグメントを繋ぎ合わせて作る
物理的には最も曲げエネルギーの小さい曲線

http://www.brnt.eu/phd/node11.html

3次Bスプライン曲線の例



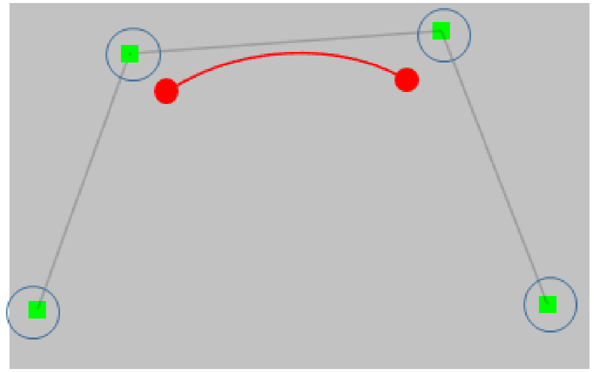




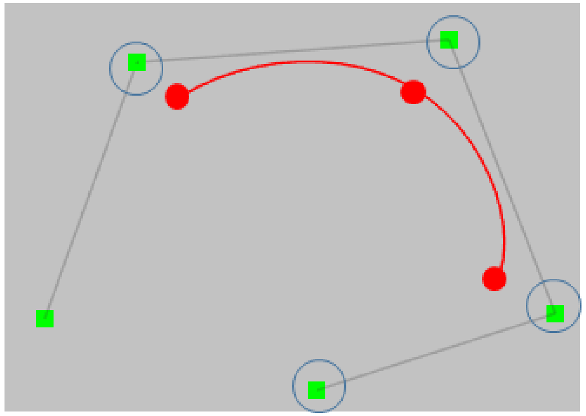
http://nurbscalculator.in/

Bスプライン曲線の特徴

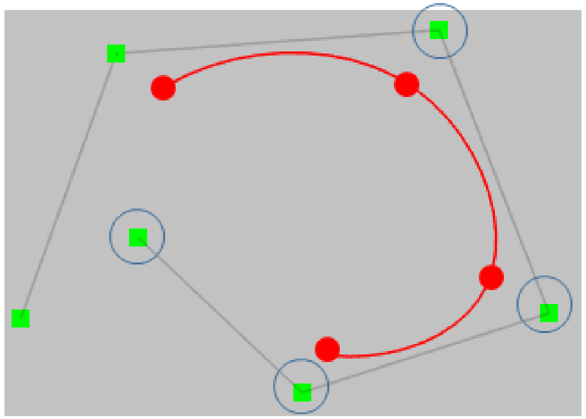
- 制御点をいくつでも指定できる。
（3次ベジェ曲線では4つだった。Bスプライン曲線では、制御点を増やすたびに新しいセグメントが追加されていく）



- セグメントが常に連続的に接続する
（ベジェ曲線ではセグメント間の連続性は保証されていなかった。）



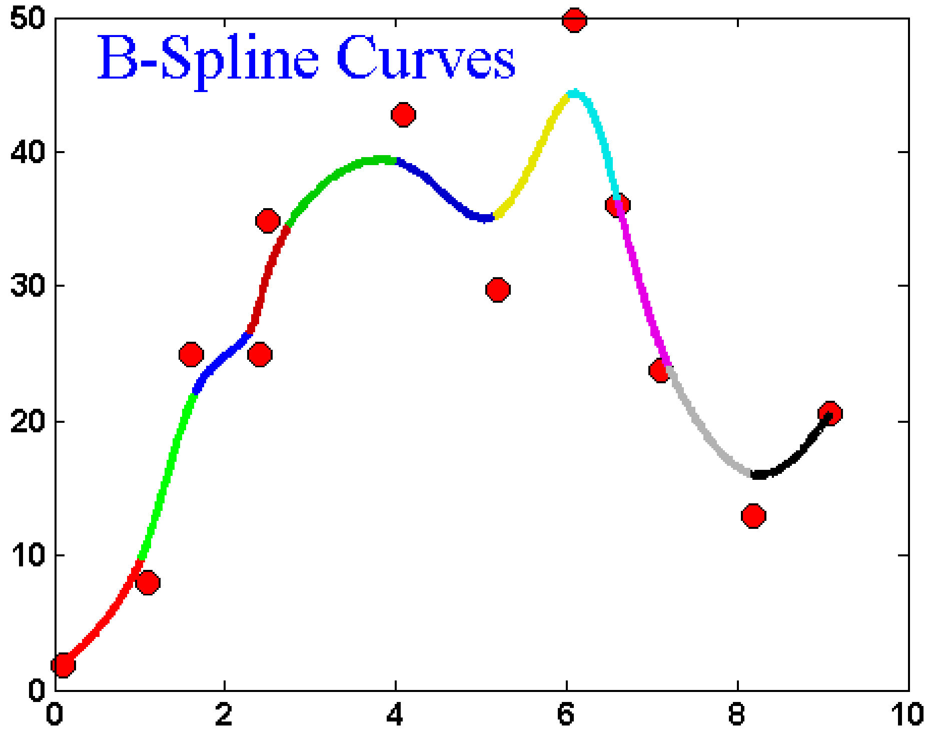
- パラメータtの値は0から1の範囲に限らない
- 局所性がある
（曲線上の1点に着目した場合、その点の位置に影響を与えるのは近傍の制御点だけ。ベジェ曲線と同じ[次数＋１]個の制御点）



- ノットベクトル（セグメントの区切りとなるパラメータの値を定義した数値（ノット）の列）で形を制御できる（ベジェ曲線を含む）

Bスプライン曲線の形を決めるもの

- 制御点列 $\mathbf{P}_i$ ＋ ノット列（ノットベクトル） t_i
- 複数の多項式曲線（セグメント）を接続して1本の曲線とする



http://profmsaeed.org/wp-content/uploads/2011/03/BSpline.html

Bスプラインの数式表現

$$\mathbf{P}(t) = \sum_{i=0}^{n+L-1} \underline{N_i^n(t)}\mathbf{P}_i$$

混合関数 「Bスプライン基底関数」

n: 次数
L: セグメントの数
制御点の数は $n+L$ （3次でセグメント数1なら制御点は4つ）

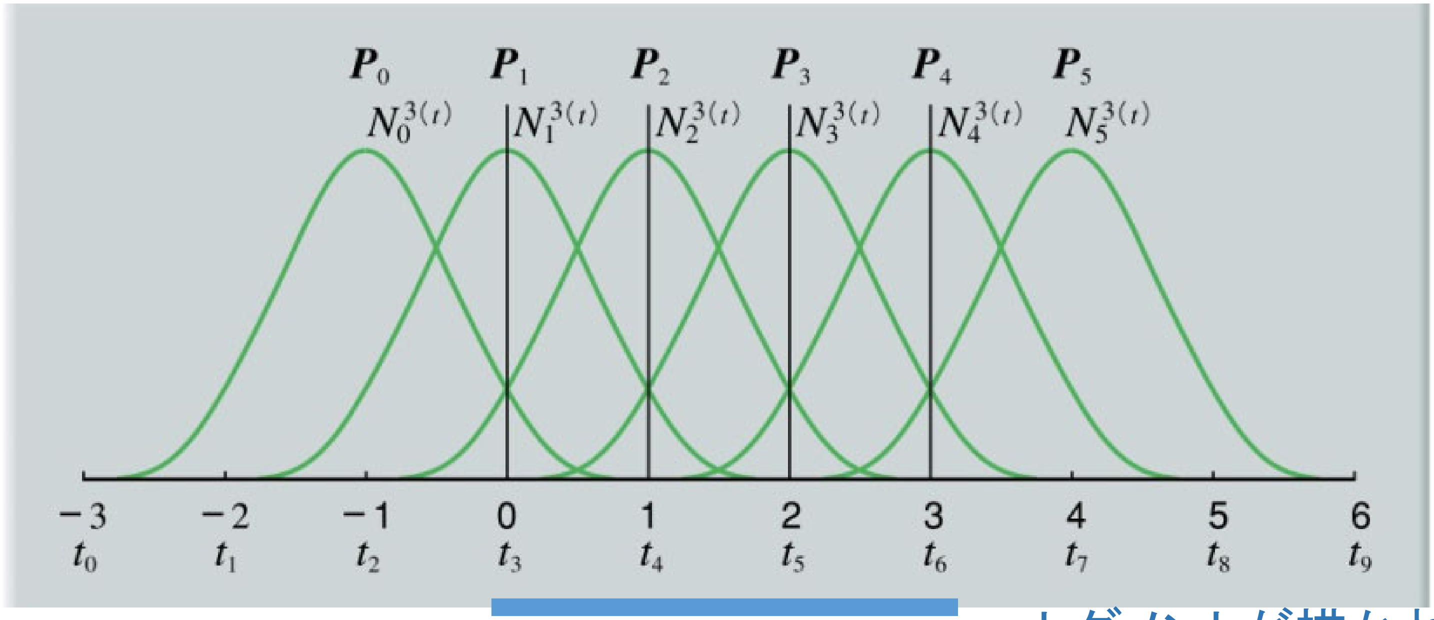
$$N_i^n(t) = \frac{t-t_i}{t_{i+n}-t_i} N_i^{n-1}(t) + \frac{t_{i+n+1}-t}{t_{i+n+1}-t_{i+1}} N_{i+1}^{n-1}(t)$$

t_i : ノット値

$$N_i^0(t) = \begin{cases} 1 & (t_i \le t < t_{i+1}) \\ 0 & \text{上記以外} \end{cases}$$

基底関数は再帰的に求まる。
手計算は大変だけどプログラムなら再帰関数ですぐ求まる。

基底関数のグラフ（一様3次）



- ノット値 ti が一定の間隔で存在する → 一様

  （ノットの間隔を変更すると「非一様」になる）
- あるtの値を見ると4つのグラフが存在する → 4つの制御点が影響
- $P(t_4)$から$P(t_5)$は、制御点P_0, P_5の影響を受けない → 局所性
- 最初と最後のノットを「次数+1」個だけ重ねる（同じ値にする）

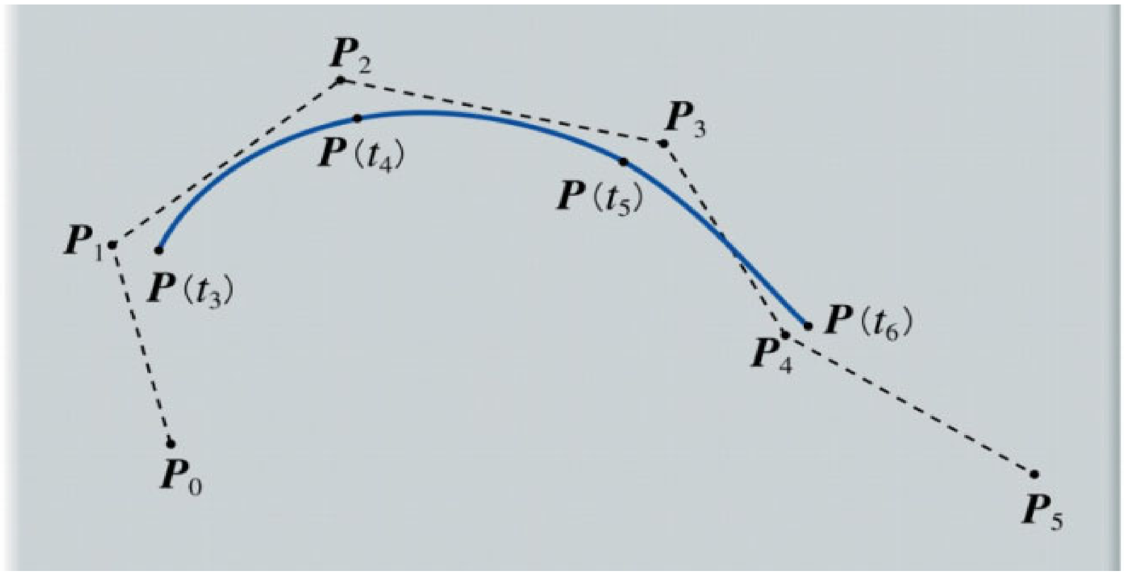
  と、端点が制御点と一致し、 ベジェ曲線と同等になる。

Bスプライン の形を決めるもの

- 「ノット列」はセグメント境界でのパラメータt の値の列。
- ノット列の値は単純増加 $t_i \leq t_{i+1}$
- パラメータ t の範囲は t_n から t_{n+L} まで。(nは次数、Lはセグメント数)
- （制御点の数）＝（次数）＋（セグメント数）
- （ノット数）＝2×（次数）+（セグメント数+1）

- 下の例は 次数3、セグメント数3、制御点数6、ノット数10

$$(t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9) = (-3, -2, -1, -0, 1, 2, 3, 4, 5, 6)$$



ベジェ曲線の表現

3次のBスプライン曲線のノット列を
（0,0,0,0,1,1,1,1）
にすると、3次ベジェ曲線と同じ曲線となる

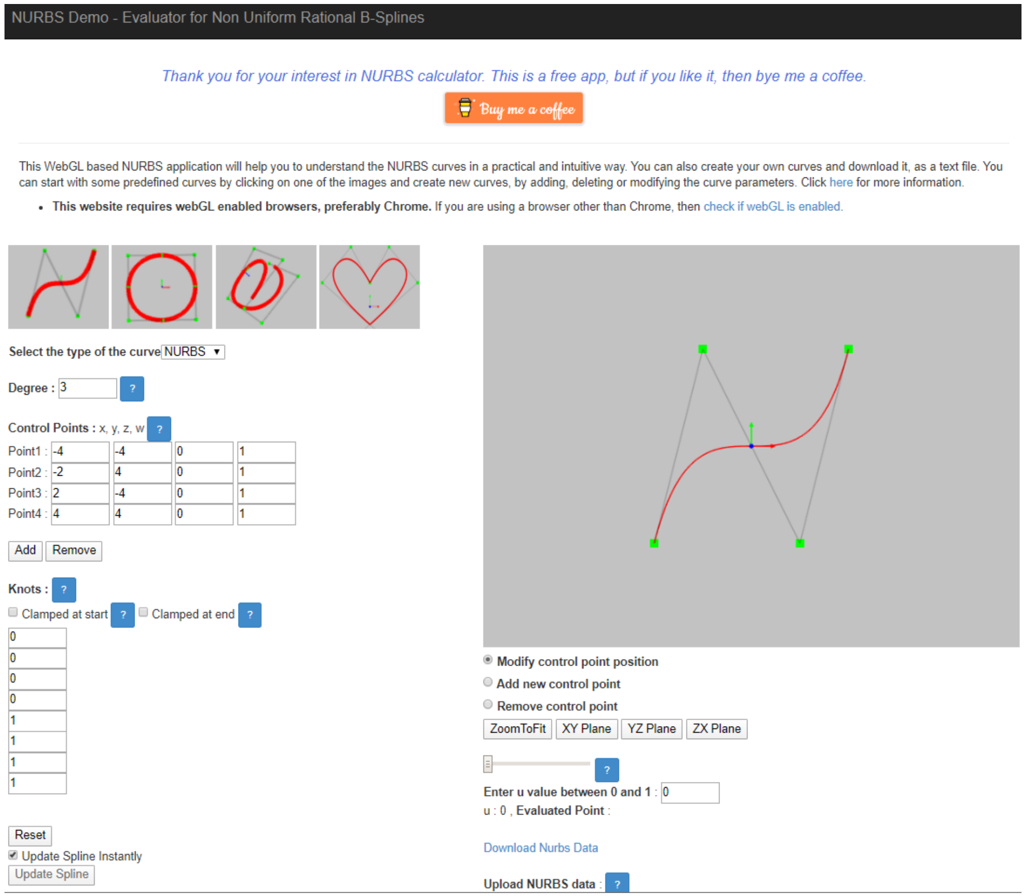
※ n次のBスプライン曲線では、n+1個のノットを重ねると、曲線は制御点を通るようになる。

練習

- 以下のWebページで Bスプライン曲線を描いてみよう

NURBS demo - WebGL based online evaluator for NURBS Curves

http://nurbscalculator.in/



Select the type of the curve B-Spline ▾

B-Spline を選択

Degree : 3 ?

3 を選択

Control Points : x, y, z, w ?

制御点の位置はマウスドラッグで操作できるので、ここは触る必要ない

Point1 :	-4	-4	0	1
Point2 :	-2	4	0	1
Point3 :	2	-4	0	1
Point4 :	4	4	0	1

制御点の移動、追加、削除

- Modify control point position
- Add new control point
- Remove control point

Knots : ?

Clamped at start ? Clamped at end ?

0
0
0
0
1
1
1
1

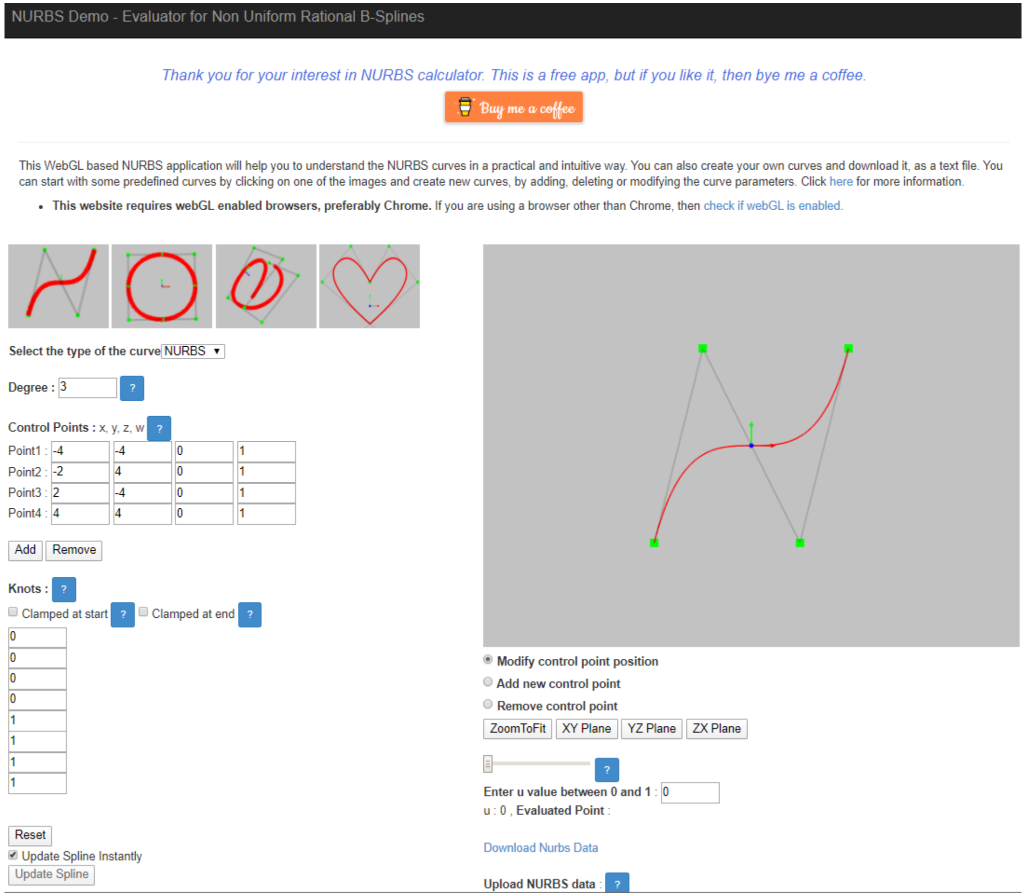
ノットベクトル。
初期状態では、0,0,0,0,1,1,1,1になっている(ベジェ曲線と同じ形になる)。

例えば、0,1,2,3,4,5,6,7 のように変更してみよう

試してみよう

- 制御点を増やしてみる。

- ノットベクトルが形状にどのように影響するか観察する。



Bスプライン曲線まとめ

- 複数のセグメントを接続して1本にしたもの
- ノット列が重要な役割をする
- 曲線の制御が局所的
- 接続の問題がない（常に滑らか）
- ベジェ曲線を含む

- ２次曲線を厳密に表現できない
  （ＮＵＲＢＳ曲線なら対応できる）

曲線の種類

- パラメトリックな自由曲線

  補間方式　　　　スプライン補間曲線

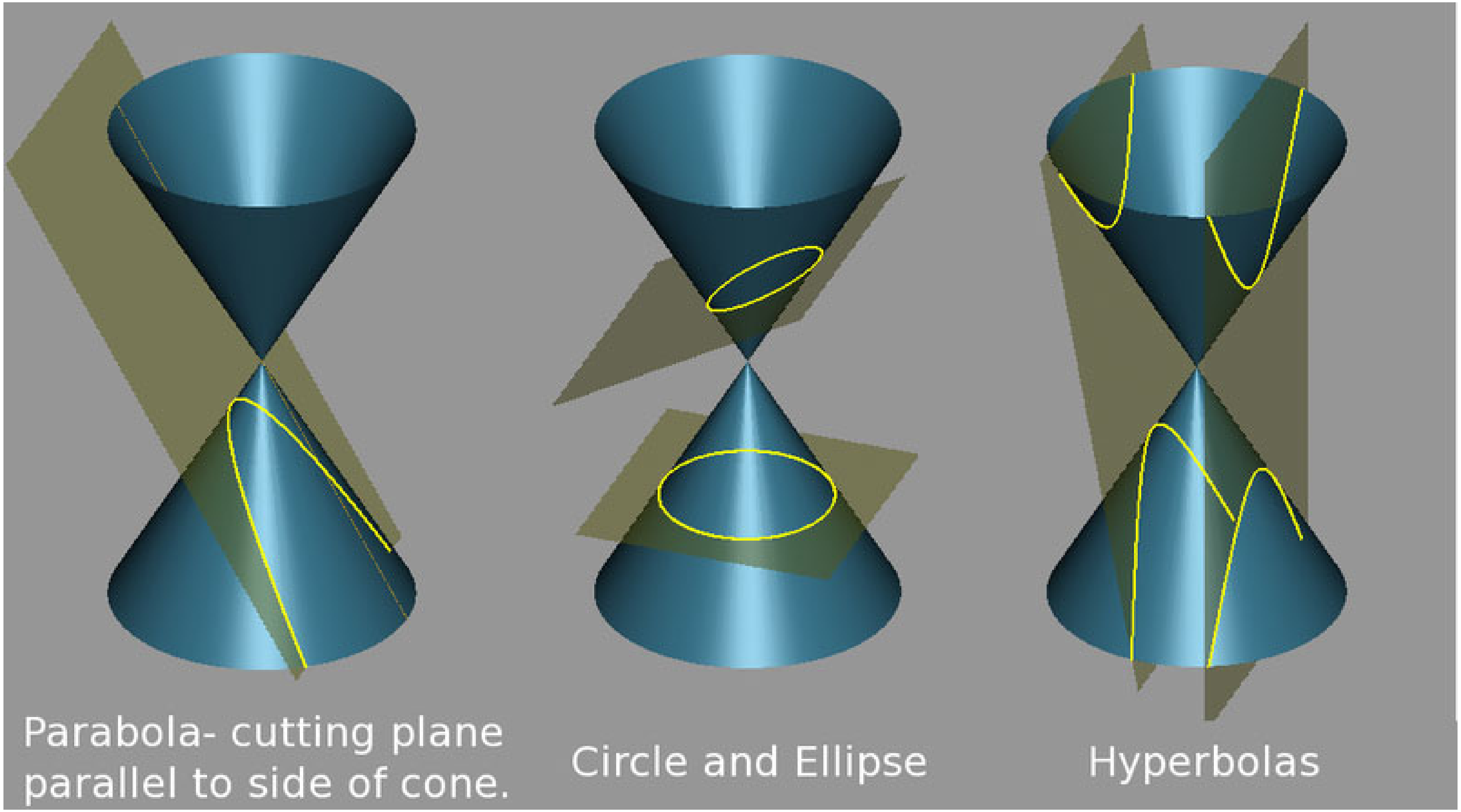
  制御点方式　　　ベジェ曲線、Ｂスプライン曲線

- 円錐曲線

  円、楕円、放物線、双曲線、（直線）

  円錐の切断によって得られるｘ、ｙの２次式

円錐曲線


Parabola- cutting plane
parallel to side of cone.
Circle and Ellipse
Hyperbolas

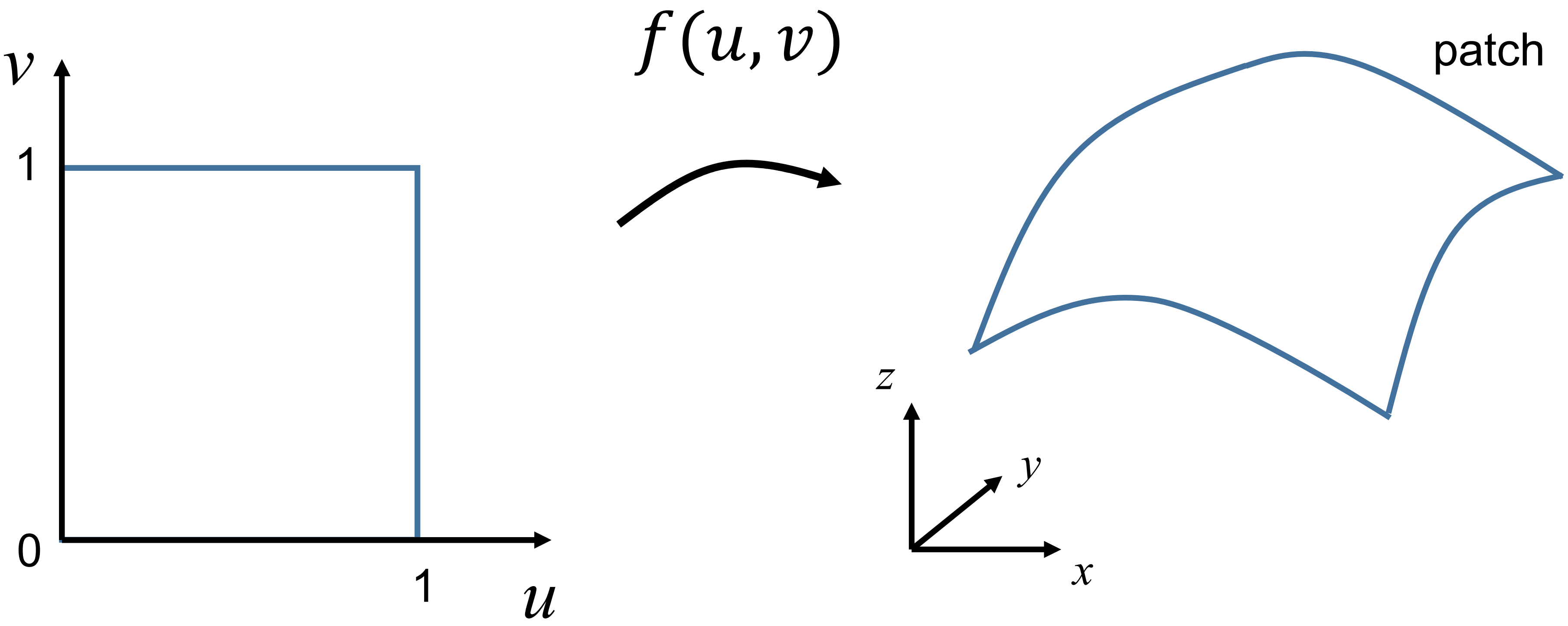
NURBS曲線

- NURBS： Non-Uniform Rational B-Spline Curve
- B-Spline 曲線を拡張したもの
- 制御点ごとに、重み wi をかけ合わせる（制御点、ノット列、重み の組み合わせで形が決まる）

- 2次曲線（円、放物線）の再現性がある

$$P(t) = \frac{\sum_{i=0}^{n+L-1} w_i P_i N_i^n(t)}{\sum_{i=0}^{n+L-1} w_i N_i^n(t)}$$

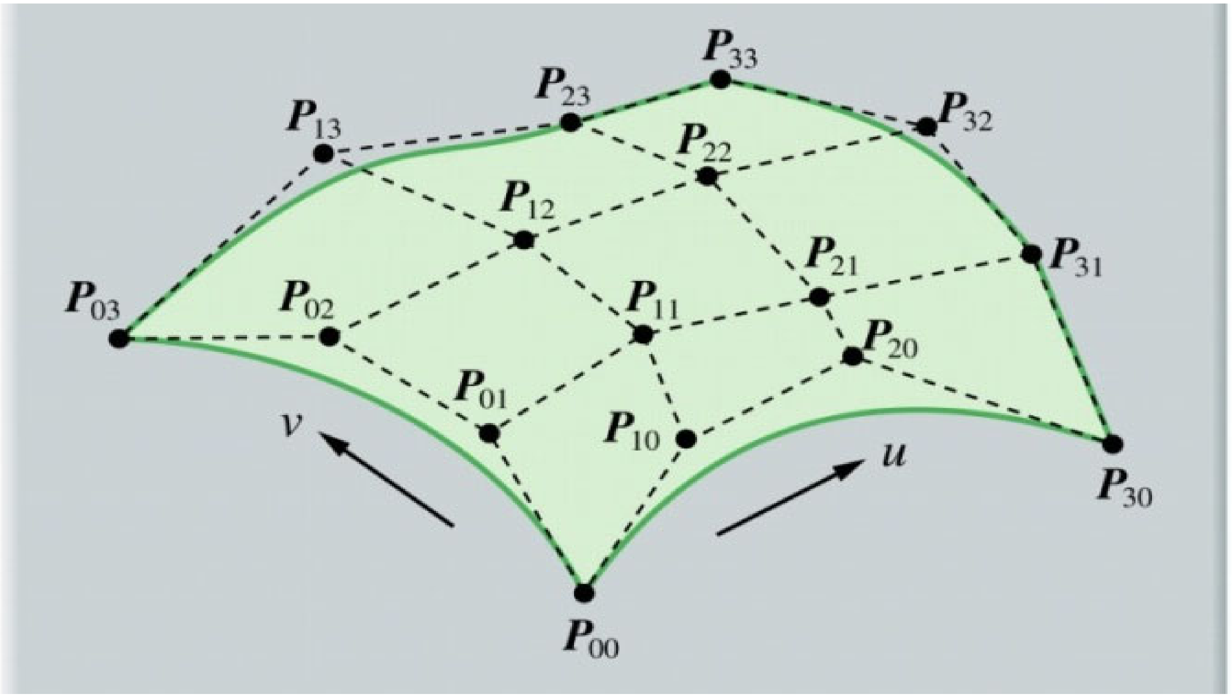
パラメトリック曲面



3 次ベジェ曲面： 双 3 次ベジェ曲面

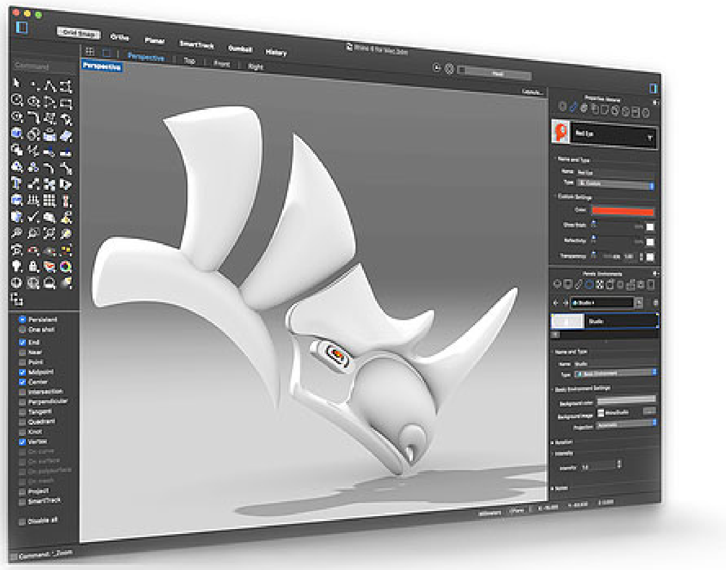
- 4×4 の格子状に並んだ 16 個の制御点 $\mathbf{P}_{ij}$ と 2 つのパラメータ u, v によって定義される。
- 4 隅の位置は制御点と一致する。

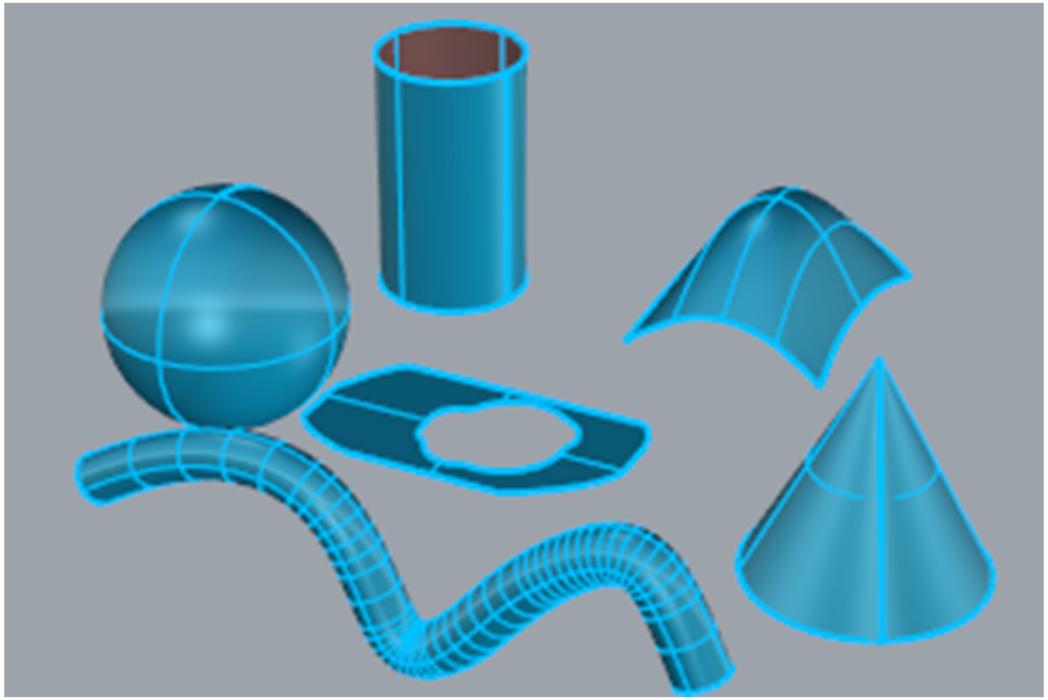
$$\mathbf{S}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} B_i^3(u) B_j^3(v) \mathbf{P}_{ij}$$



NURBSを基本としたCGソフトウェア

Rhinoceros®
modeling tool for designers





課題

Bスプライン曲線を描画する