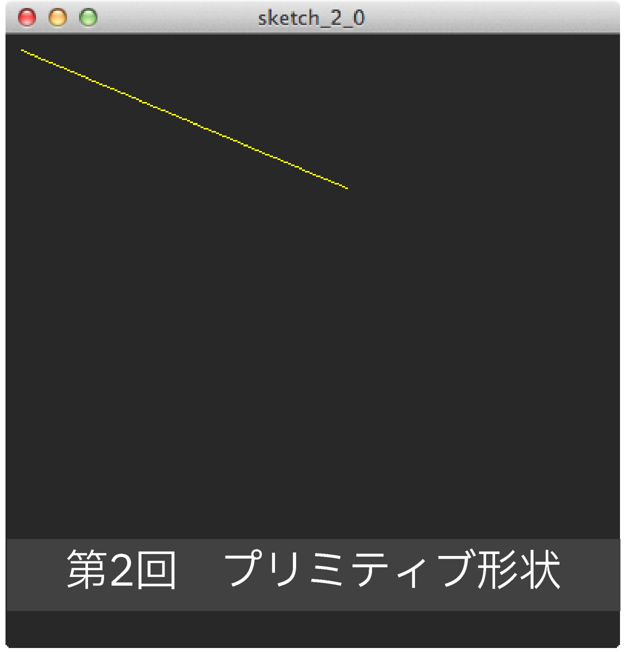
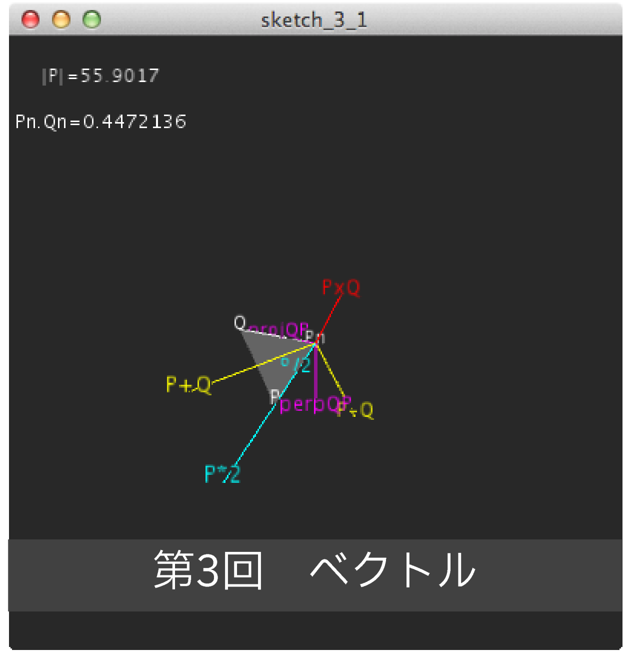
CGとCADの数理

GEOMETRIC MODELING AND COMPUTER GRAPHICS

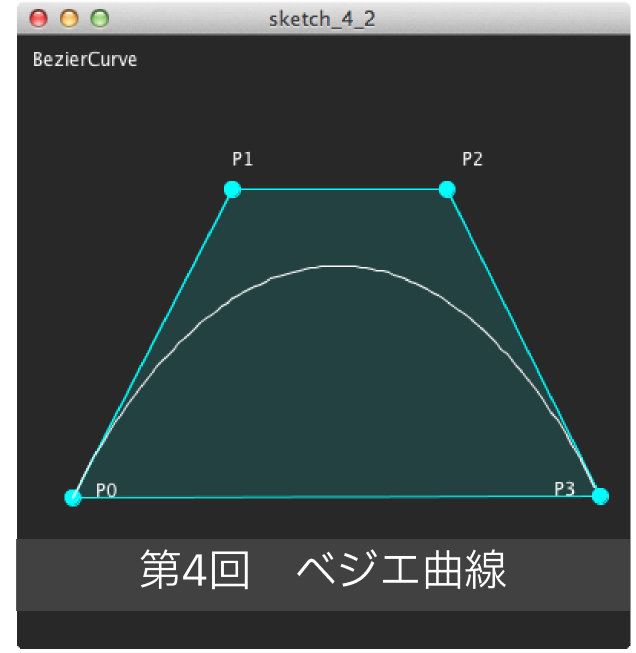
第04回 ベジエ曲線



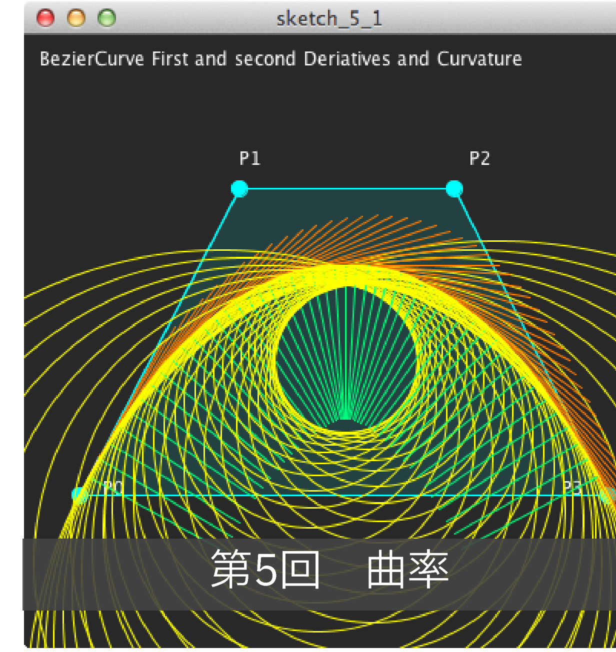
第2回　プリミティブ形状



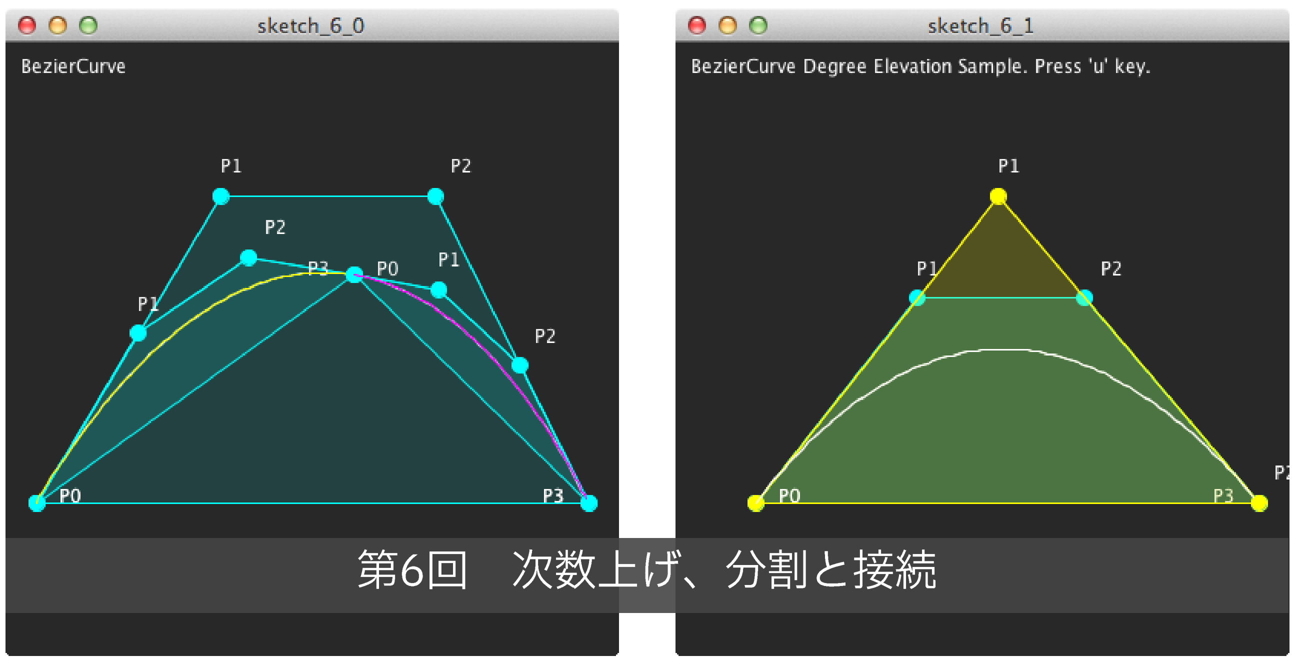
第3回　ベクトル



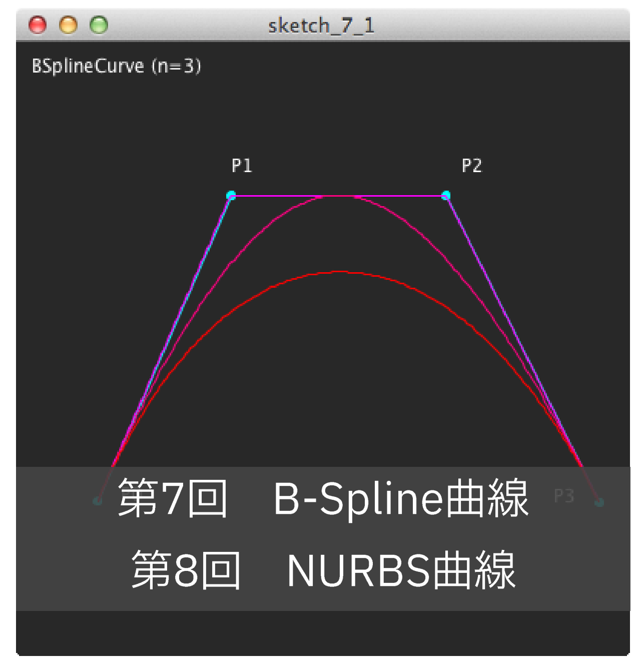
第4回　ベジエ曲線



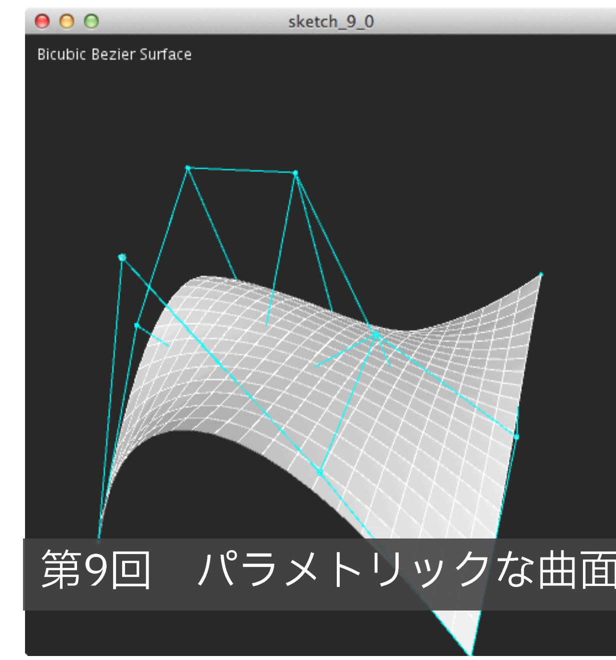
第5回　曲率



第6回　次数上げ、分割と接続



第7回　B-Spline曲線
第8回　NURBS曲線

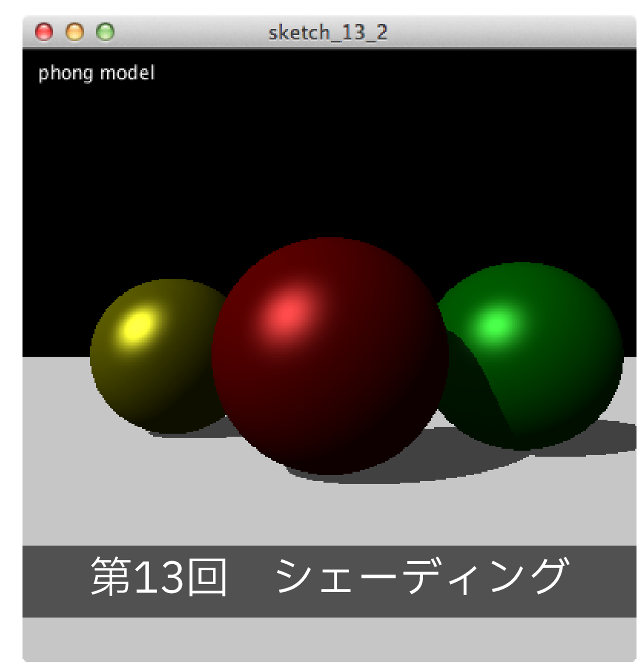


第9回　パラメトリックな曲面

第10回　行列

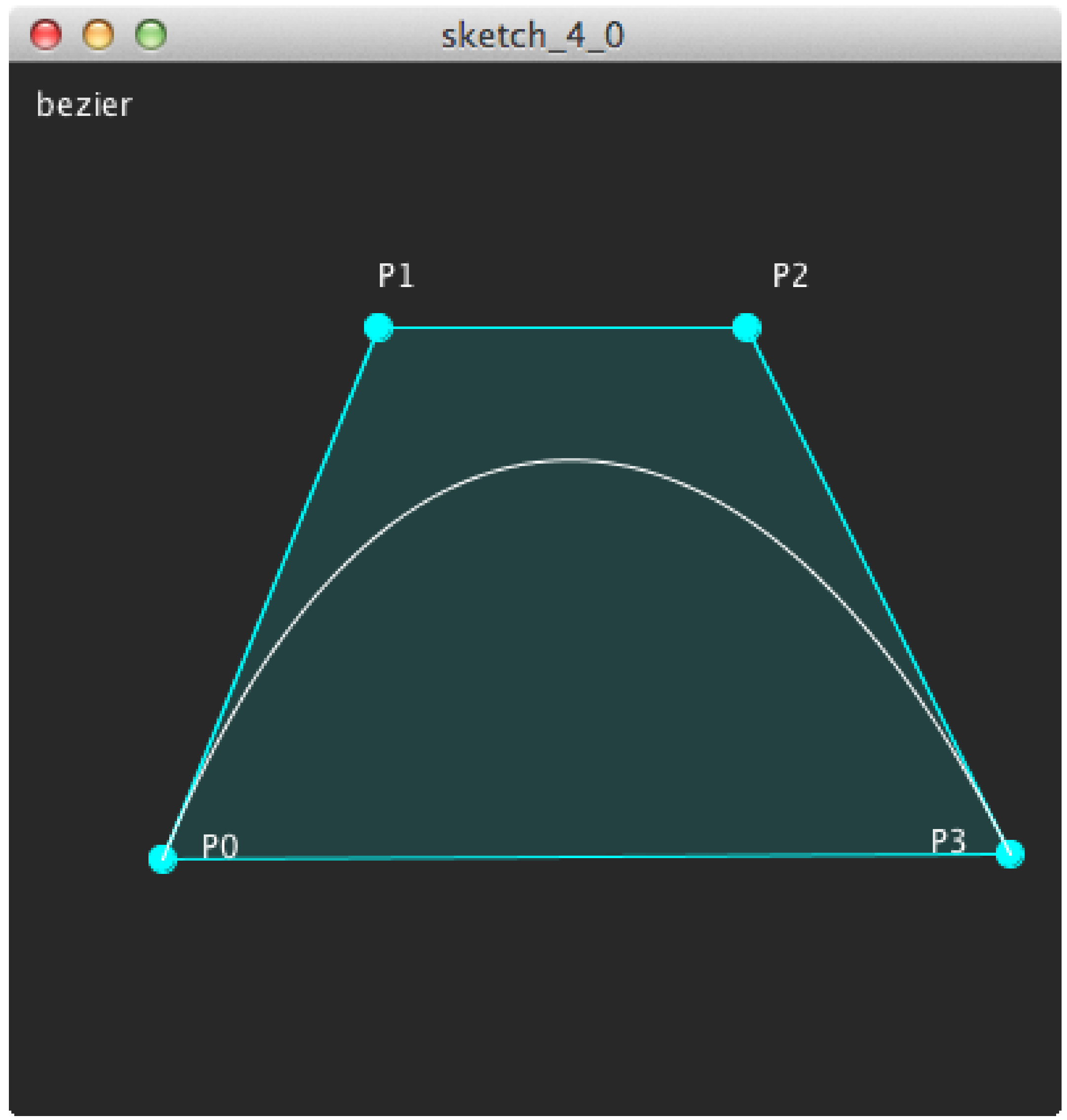
第11回　三次元物体の表示

第12回　レイトレーシング



第13回　シェーディング

ベジエ曲線
Bezier Curve

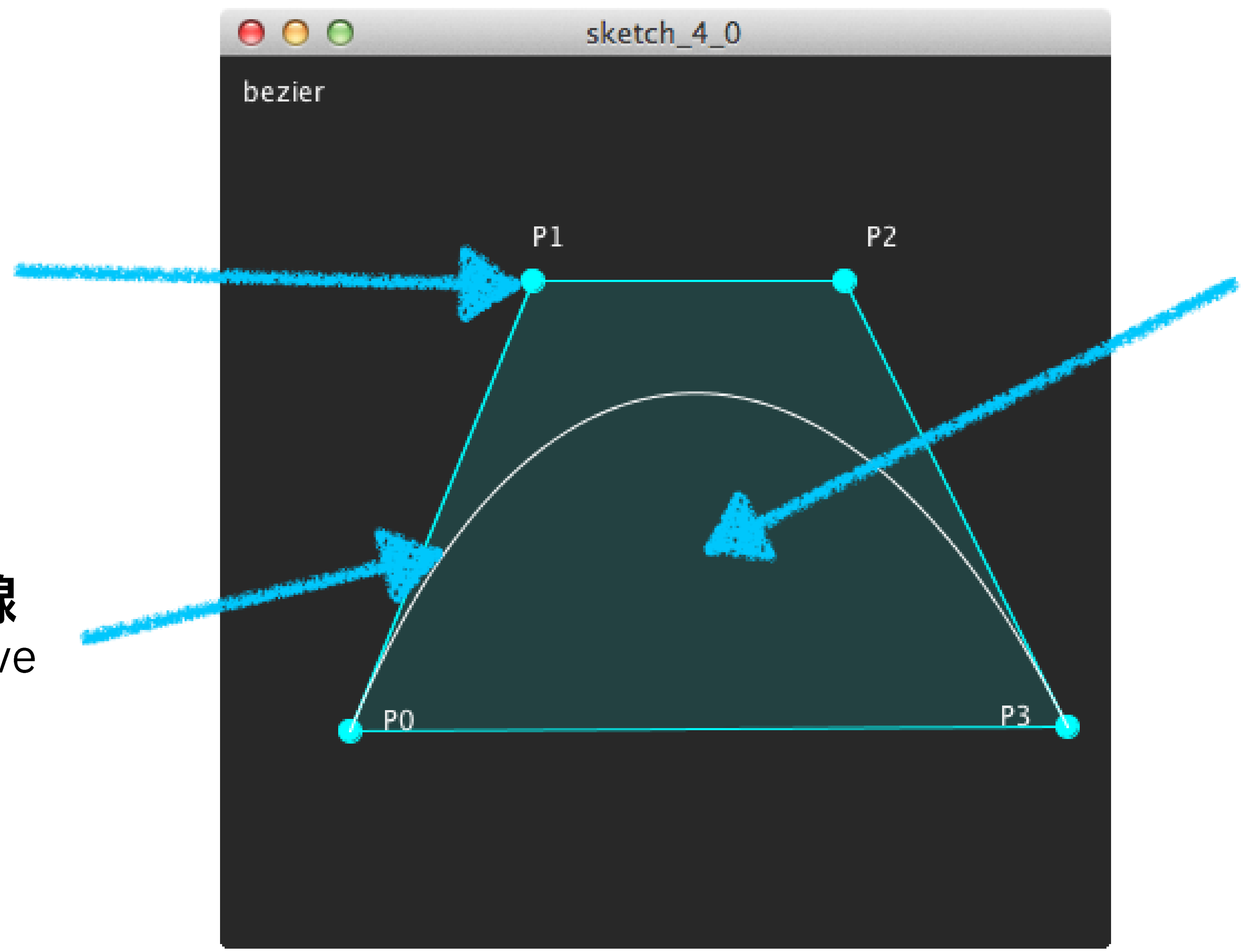


CGでは滑らかな曲線を描く場合に良く利用されます。フランスの自動車メーカーであるシトロエン社のド・カステリョ（Paul de Casteliau)とピエール・ベジエ（Pierre Bézier）によって考案されました。本日は３次ベジエ曲線（Cubic Bezier Curve）のアルゴリズムを理解しましょう。

制御点と制御ポリゴン
Control Point and Control Polygon

制御点
Control Point

3次ベジエ曲線
Cubic Bezier Curve



制御ポリゴン
Control Polygon

３次ベジエ曲線は、４つの制御点からなります。**ベジエ曲線では必ず、「制御点の数 - 1」次数のベジエ曲線が生成されます。**また、**生成されたベジエ曲線は必ず制御ポリゴンの内側にあります。**これを**凸包性**と言います。ちなみに、2次ベジエ曲線はQuadratic Bezier Curve と言います。

SOLからsketch_4_0.zip をダウンロードして下さい

```
void draw() {
  background(40);

  p0.x = mouseX;  p0.y = mouseY;

  // line
  stroke(0, 255, 255);
  fill(0, 255, 255, 30);
  quad(p0.x, p0.y, p1.x, p1.y, p2.x, p2.y, p3.x, p3.y);
  line(p0.x, p0.y, p1.x, p1.y); // p0 - p1
  line(p1.x, p1.y, p2.x, p2.y); // p1 - p2
  line(p2.x, p2.y, p3.x, p3.y); // p2 - p3

  // draw control points
  fill(0, 255, 255);
  ellipse(p0.x, p0.y, 10, 10);   // p0
  ellipse(p1.x, p1.y, 10, 10);   // p1
  ellipse(p2.x, p2.y, 10, 10);   // p2
  ellipse(p3.x, p3.y, 10, 10);   // p3

  // text control points
  fill(255, 255, 255);
  text("P0", p0.x+15, p0.y    ); // p0
  text("P1", p1.x,    p1.y-15); // p1
  text("P2", p2.x+10, p2.y-15); // p2
  text("P3", p3.x-30, p3.y    ); // p3

  // draw bezier curve
  noFill();
  stroke(255, 255, 255);

  text("bezier", 10, 20);

}
```

```
bezier(p0.x, p0.y, p1.x, p1.y, p2.x, p2.y, p3.x, p3.y);
```

P_0 P_1 P_2 P_3

頂点 P_0 は、カーソルに追従させます。

制御ポリゴンを描画しています。

quad(P_0のX座標, P_0のY座標, P_1のX座標, P_1のY座標,
P_2のX座標, P_2のY座標, P_3のX座標, P_3のY座標);

で矩形が描画できます。

line(始点のX座標, 始点のY座標
終点のX座標, 終点のY座標);

で稜線が描画できます。

制御点 P_0、P_1、P_2、P_3の４つを描画しています。

制御点 P_0、P_1、P_2、P_3の傍にテキスト表示ます。

ベルンシュタイン既定関数

Bernstein Basis Function

ベジエ曲線は、以下の式で計算されます。

n次→(n+1)個の制御点

制御点

$$R(t) = \sum_{i=0}^{n} B_i^n(t)\, P_i$$

ただし、$0 \le t \le 1$

パラメーター tの範囲

パラメーター t

このようにパラメーターで表されている曲線をパラメトリック曲線と言います。

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

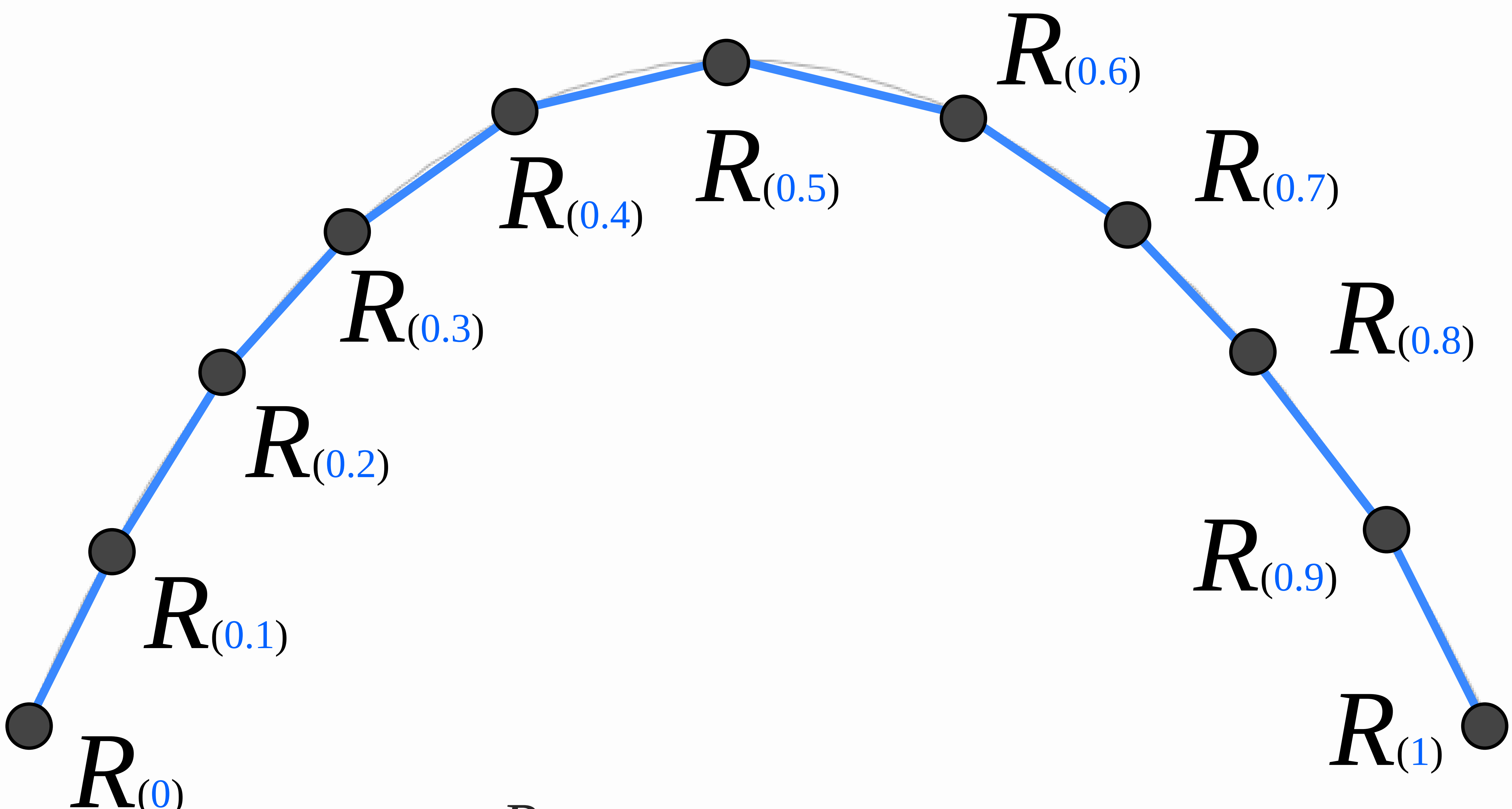
$_nC_i$ と同じです。

制御点をP_n、ベジエ曲線上の点を$R_{(t)}$とします。

この B_i^n を **Bernstein Basis Function**（ベルンシュタイン基底関数）と言います。

ベジエ曲線のイメージ

$$R(t) = \sum_{i=0}^{n} B_i^n(t)\, P_i$$ ただし、$0 \leq t \leq 1$



ベジエ曲線上の点を$R_{(t)}$をたくさん繋ぎ合わせれば、曲線を近似できます。

3次ベジエ曲線
Cubic Bezier Curve

今回は、3次ベジエ曲線です。次数は、$n=3$ ですので、それを書き下してみましょう。

次数 $n=3$

制御点

$$R(t) = \sum_{i=0}^{n} B_i^n(t)\, P_i$$

ただし、$0 \le t \le 1$

パラメーター tの範囲

パラメーター t

このようにパラメーターで表されている曲線をパラメトリック曲線と言います。

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

$$R(t) = \sum_{i=0}^{3} B_i^3(t)\, P_i$$

ただし、$0 \le t \le 1$

$$R(t) = \sum_{i=0}^{3} B_i^3(t)\, P_i \qquad B_i^n = \binom{n}{i}(1-t)^{n-i}\, t^i$$

$i=0$ のとき → $B_0^3(t)P_0$ → $\binom{3}{0}(1-t)^{3-0}\, t^0 = (1-t)^3$, $\binom{3}{0} = 1$

$i=1$ のとき → $B_1^3(t)P_1$ → $\binom{3}{1}(1-t)^{3-1}\, t^1 = 3(1-t)^2 t$, $\binom{3}{1} = 3$

$i=2$ のとき → $B_2^3(t)P_2$ → $\binom{3}{2}(1-t)^{3-2}\, t^2 = 3(1-t)\, t^2$, $\binom{3}{2} = 3$

$i=3$ のとき → $B_3^3(t)P_3$ → $\binom{3}{3}(1-t)^{3-3}\, t^3 = t^3$, $\binom{3}{3} = 1$

$$R(t) = \sum_{i=0}^{3} B_i^3(t)\, P_i$$

$$R(t) = B_0^3(t)P_0 + B_1^3(t)P_1 + B_2^3(t)P_2 + B_3^3(t)P_3$$

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

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

具体的にどうコーディングすればいいの？

$$R(t) = (1 - t)^3 P_0 + 3(1 - t)^2 t\, P_1 + 3(1 - t)\, t^2 P_2 + t^3 P_3$$

$$P_1 = (P_{1.x}, P_{1.y}) \qquad P_3 = (P_{3.x}, P_{3.y})$$

$$P_0 = (P_{0.x}, P_{0.y}) \qquad P_2 = (P_{2.x}, P_{2.y})$$

$$R(t).x = (1 - t)^3 P_{0.x} + 3(1 - t)^2 t\, P_{1.x} + 3(1 - t)\, t^2 P_{2.x} + t^3 P_{3.x}$$

$$R(t).y = (1 - t)^3 P_{0.y} + 3(1 - t)^2 t\, P_{1.y} + 3(1 - t)\, t^2 P_{2.y} + t^3 P_{3.y}$$

```
P0 = new PVector(); P0.x =  20; P0.y = 300;
P1 = new PVector(); P1.x = 140; P1.y = 100;
P2 = new PVector(); P2.x = 280; P2.y = 100;
P3 = new PVector(); P3.x = 380; P3.y = 300;
```

P_0、P_1、P_2、P_3のXY座標はプログラム済みなので、後は代入で計算できます。

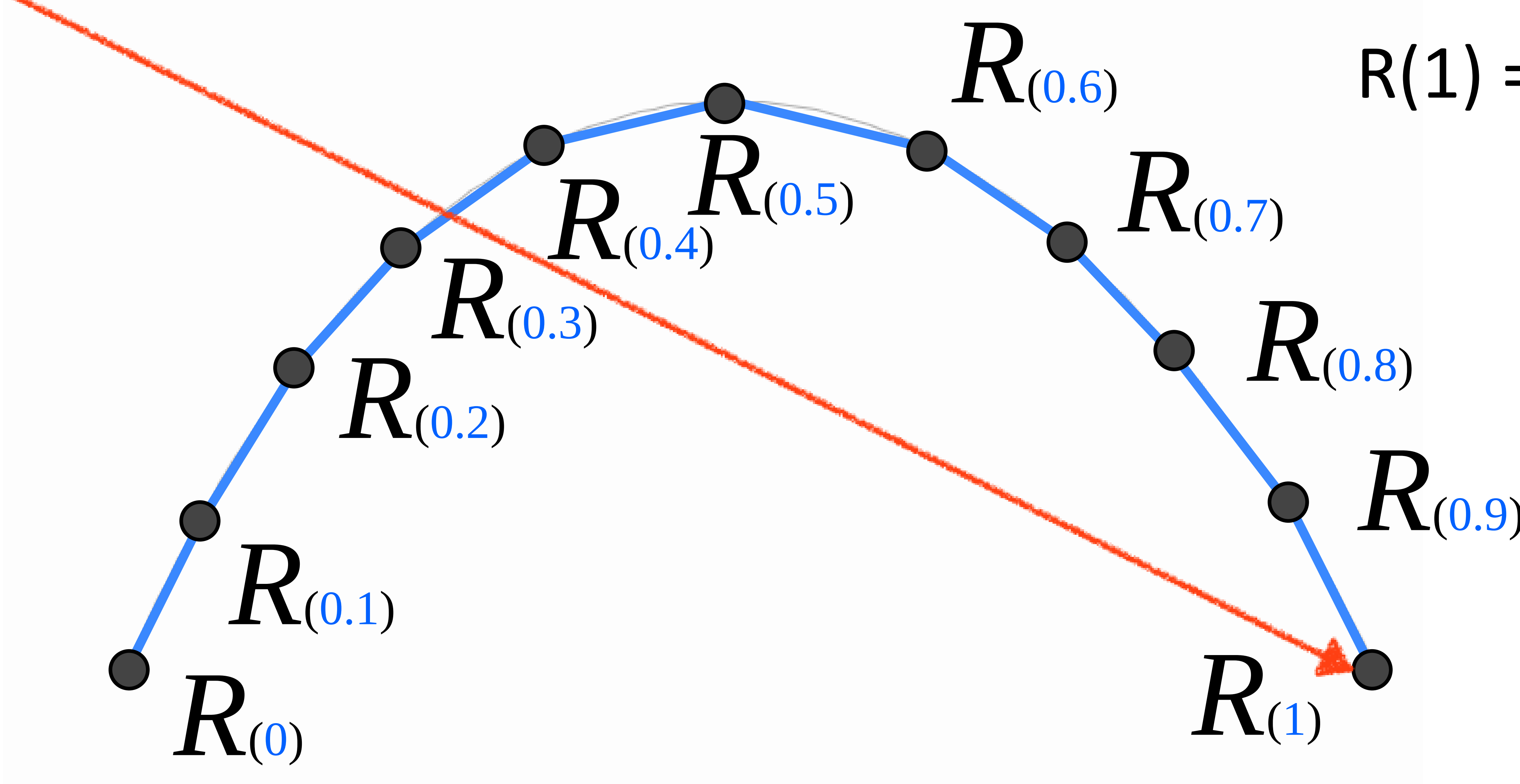
$$R(t) = (1-t)^3 P_0 + 3(1-t)^2 t\, P_1 + 3(1-t)\, t^2 P_2 + t^3 P_3$$

$$R(t).x = (1-t)^3 P_{0.x} + 3(1-t)^2 t\, P_{1.x} + 3(1-t)\, t^2 P_{2.x} + t^3 P_{3.x}$$
$$R(t).y = (1-t)^3 P_{0.y} + 3(1-t)^2 t\, P_{1.y} + 3(1-t)\, t^2 P_{2.y} + t^3 P_{3.y}$$

t=1を代入！

R(0) = P(0)

R(1) = P(3)



sketch_4_1.zip 開いてください。

$$R(t) = \sum_{i=0}^{3} B_i^3(t)\ P_i \qquad B_i^n = \binom{n}{i}(1-t)^{n-i}\,t^i$$

$i=0$ のとき → $\boxed{B_0^3(t)P_0}\ \boxed{=(1-t)^3}$

```
B30t = (1-t)*(1-t)*(1-t);
```

$i=1$ のとき → $\boxed{B_1^3(t)P_1}\ \boxed{=3(1-t)^2\,t}$

```
B31t = 3*(1-t)*(1-t)*t;
```

$i=2$ のとき → $\boxed{B_2^3(t)P_2}\ \boxed{=3(1-t)\,t^2}$

```
B32t = 3*(1-t)*t*t;
```

$i=3$ のとき → $\boxed{B_3^3(t)P_3}\ \boxed{=t^3}$

```
B33t = t*t*t;
```

$$R(t) = \sum_{i=0}^{3} B_i^3(t)\, P_i \qquad B_i^n = \binom{n}{i}(1-t)^{n-i}\, t^i$$

$i=0$ のとき → `B30t = (1-t)*(1-t)*(1-t);`

$i=1$ のとき → `B31t = 3*(1-t)*(1-t)*t;`

$i=2$ のとき → `B32t = 3*(1-t)*t*t;`

$i=3$ のとき → `B33t = t*t*t;`

$\boldsymbol{x}$ に関して

$$R(t).x = \boxed{(1-t)^3} P_{0.x} + \boxed{3(1-t)^2\, t} P_{1.x} + \boxed{3(1-t)\, t^2} P_{2.x} + \boxed{t^3} P_{3.x}$$

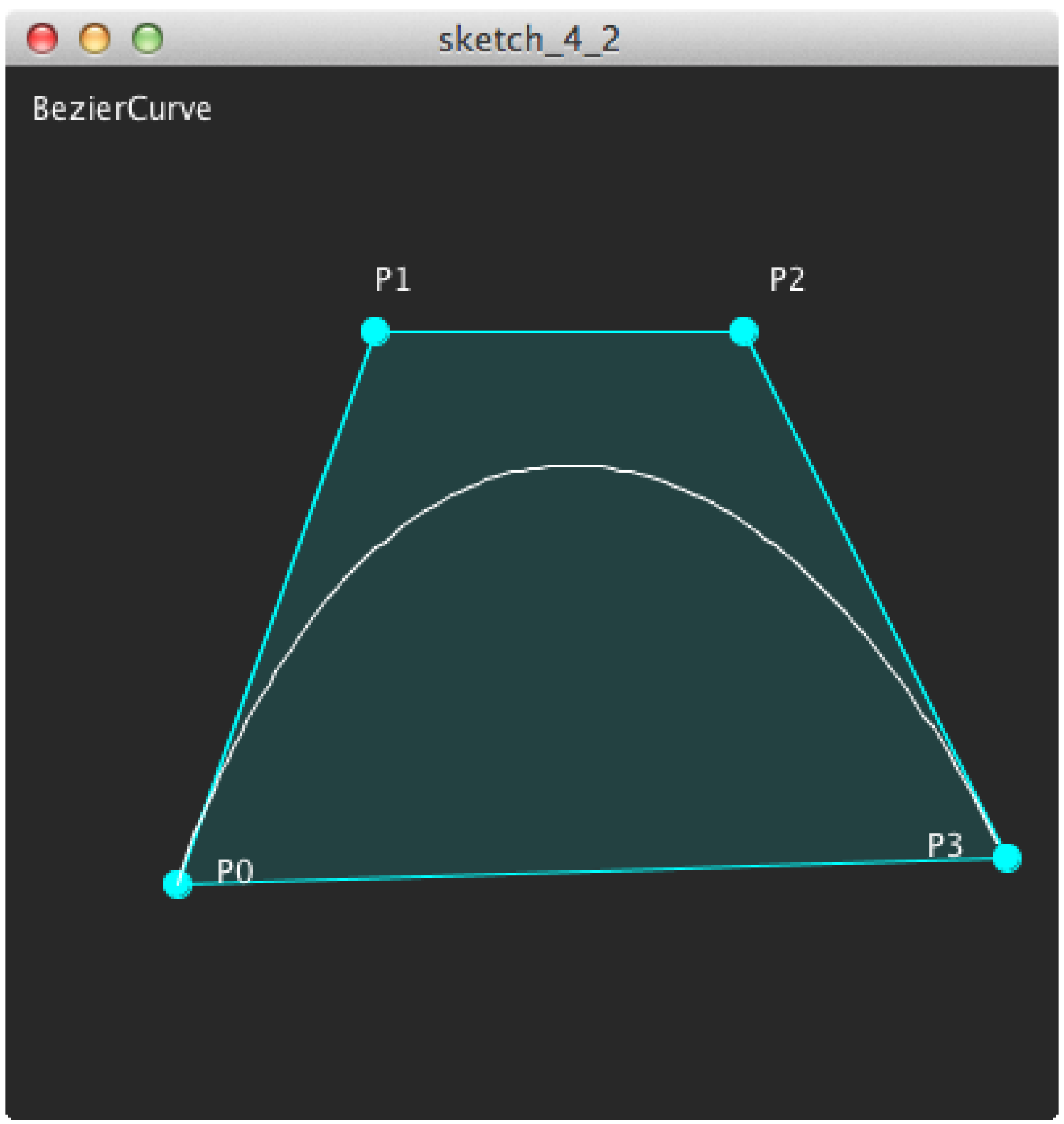
```
R[tt].x = B30t *P0.x + B31t *P1.x + B32t *P2.x + B33t *P3.x
```

$\boldsymbol{y}$ に関して

$$R(t).y = \boxed{(1-t)^3} P_{0.y} + \boxed{3(1-t)^2\, t} P_{1.y} + \boxed{3(1-t)\, t^2} P_{2.y} + \boxed{t^3} P_{3.y}$$

```
R[tt].y = B30t *P0.y + B31t *P1.y + B32t *P2.y + B33t *P3.y
```

```
for(tt = 0; tt < tn ; tt+=1) {
    B30t =     (1-t)*(1-t)*(1-t)          ;
    B31t = 3 * (1-t)*(1-t)        *t      ;
    B32t = 3 * (1-t)              *t*t    ;
    B33t =                         t*t*t  ;
    R[tt] = new PVector();
    R[tt].x = B30t*P0.x + B31t*P1.x + B32t*P2.x + B33t*P3.x;
    R[tt].y = B30t*P0.y + B31t*P1.y + B32t*P2.y + B33t*P3.y;
  if (tt != 0) line(R[tt-1].x, R[tt-1].y, R[tt].x, R[tt].y);
  t = t + ts;
}
```



1. 実行できた方は、カーソルを動かして遊んでみて下さい。

2. tn = 100; の部分が曲線の分割に関係しています。試しに、tn = 4 など、好きな整数を入れて遊んでみて下さい。