

コンピュータグラフィクス論

－モデリング(1)－

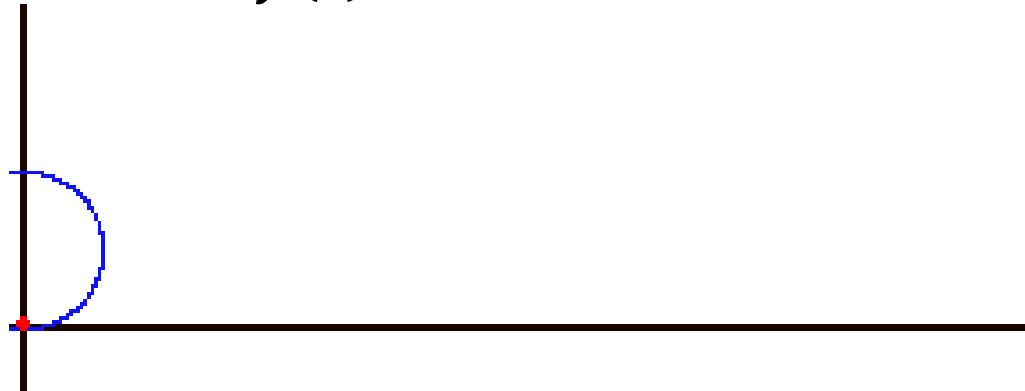
2017年4月13日

高山 健志

パラメトリック曲線

- X座標とY座標がパラメタ t (\equiv 時刻) によって決まるもの
 - 例：サイクロイド

$$\begin{aligned}x(t) &= t - \sin t \\y(t) &= 1 - \cos t\end{aligned}$$



- 接線ベクトル： $(x'(t), y'(t))$
- 多項式曲線： $x(t) = \sum_i a_i t^i$

3次エルミート曲線

- 両端での値と微分の制約を満たす
(=エルミート補間) ような、多項式曲線

- 制約が4つなので、4自由度が必要

→ 4個の係数 → 3次多項式

- $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$
- $x'(t) = a_1 + 2a_2 t + 3a_3 t^2$

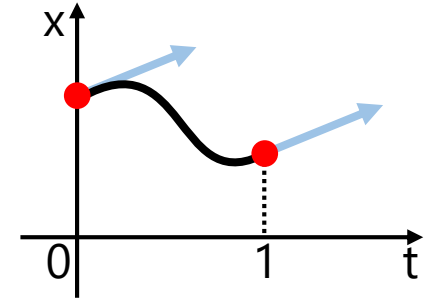
- 制約値を代入すれば係数が求まる

$$x(0) = x_0$$

$$x(1) = x_1$$

$$x'(0) = x'_0$$

$$x'(1) = x'_1$$



$$x(0) = a_0 = x_0$$

$$x(1) = a_0 + a_1 + a_2 + a_3 = x_1$$

$$x'(0) = a_1 = x'_0$$

$$x'(1) = a_1 + 2a_2 + 3a_3 = x'_1$$

→

$$a_0 = x_0$$

$$a_1 = x'_0$$

$$a_2 = -3x_0 + 3x_1 - 2x'_0 - x'_1$$

$$a_3 = 2x_0 - 2x_1 + x'_0 + x'_1$$

ベジエ (Bezier[ベズィエ]) 曲線

- 入力：3点 P_0, P_1, P_2 (制御点)
 - 任意空間 (2D, 3D) の座標

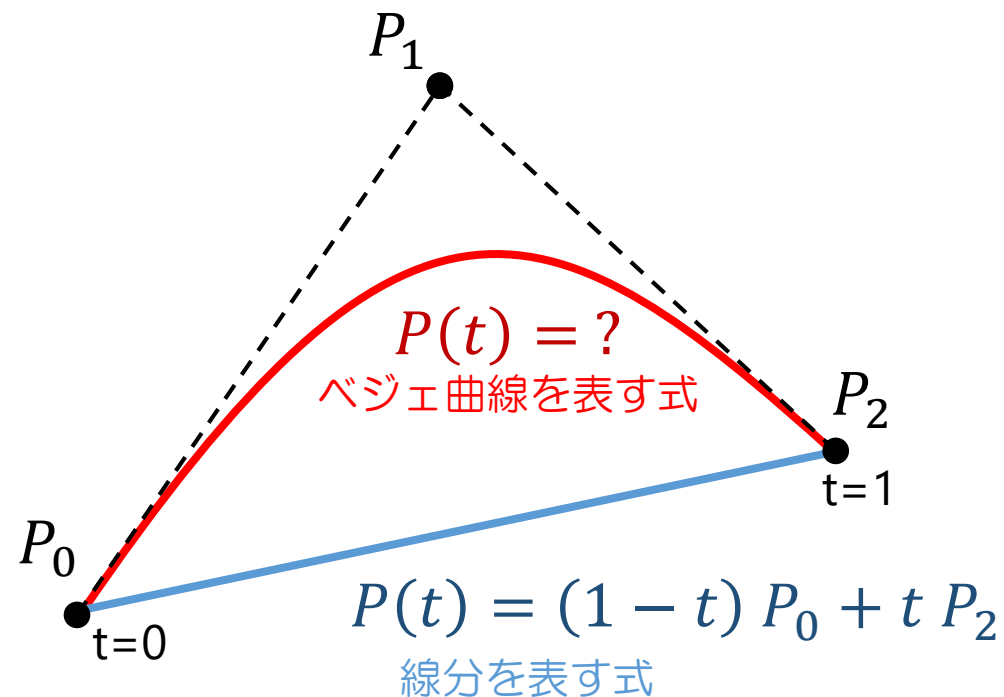
- 求めたいもの：曲線 $P(t)$ で

$$P(0) = P_0$$

$$P(1) = P_2$$

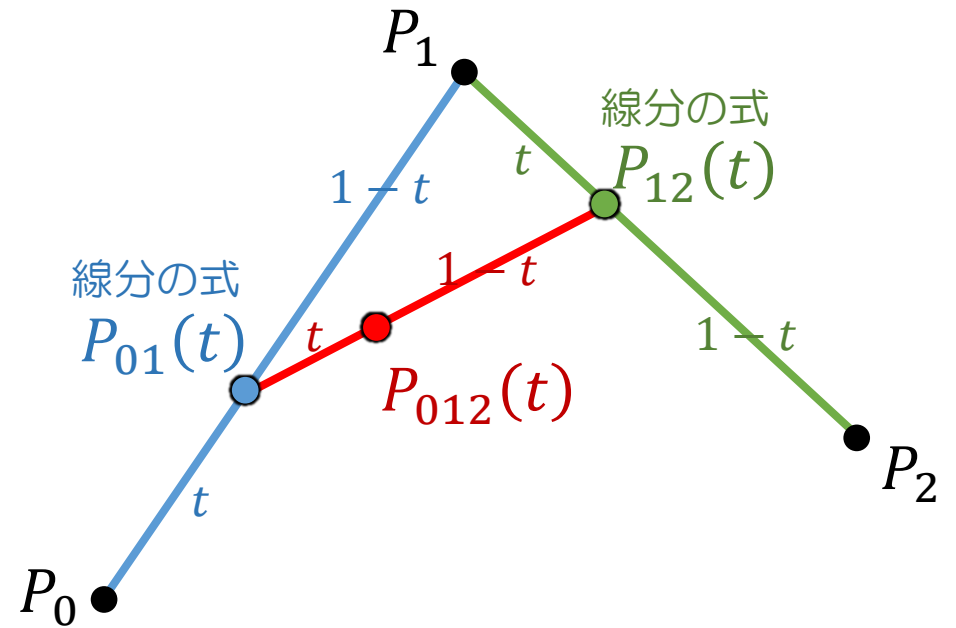
を満たしつつ、

P_1 に“引っ張られる”もの



ベジエ曲線

- $P_{01}(t) = (1-t)P_0 + tP_1$
- $P_{12}(t) = (1-t)P_1 + tP_2$
 - $P_{01}(0) = P_0$
 - $P_{12}(1) = P_2$



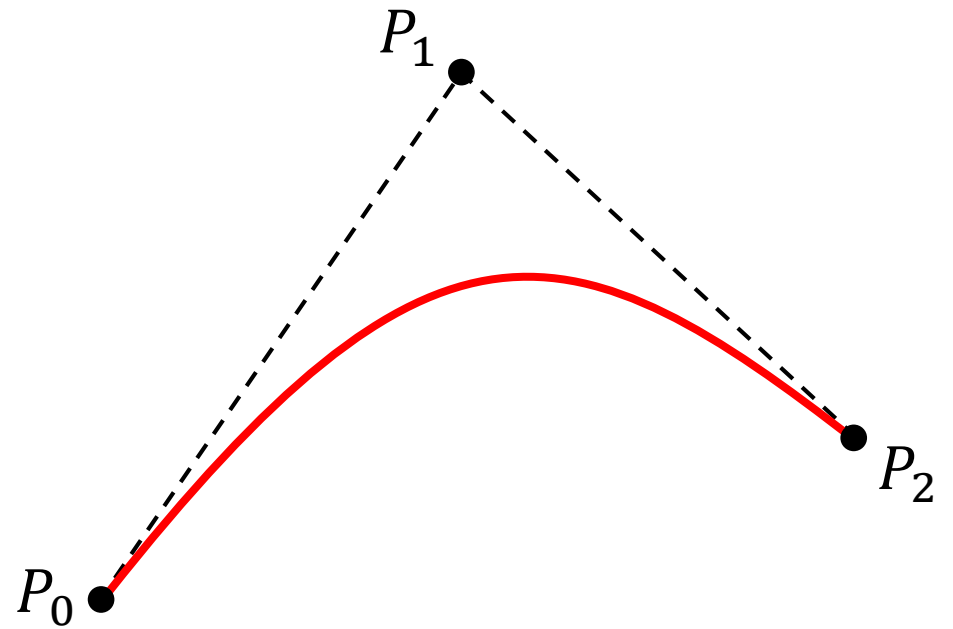
- アイディア：“補間を補間”
 $t: 0 \rightarrow 1$ のとき $P_{01} \rightarrow P_{12}$ となるように線形補間

- $P_{012}(t) = (1-t)P_{01}(t) + tP_{12}(t)$
 $= (1-t)\{(1-t)P_0 + tP_1\} + t\{(1-t)P_1 + tP_2\}$
 $= (1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2$

2次ベジエ曲線

ベジエ曲線

- $P_{01}(t) = (1 - t)P_0 + t P_1$
- $P_{12}(t) = (1 - t)P_1 + t P_2$
 - $P_{01}(0) = P_0$
 - $P_{12}(1) = P_2$



- アイディア：“補間を補間”
 $t : 0 \rightarrow 1$ のとき $P_{01} \rightarrow P_{12}$ となるように線形補間

- $P_{012}(t) = (1 - t)P_{01}(t) + t P_{12}(t)$
$$= (1 - t)\{(1 - t)P_0 + t P_1\} + t \{(1 - t)P_1 + t P_2\}$$
$$= (1 - t)^2 P_0 + 2t(1 - t)P_1 + t^2 P_2$$

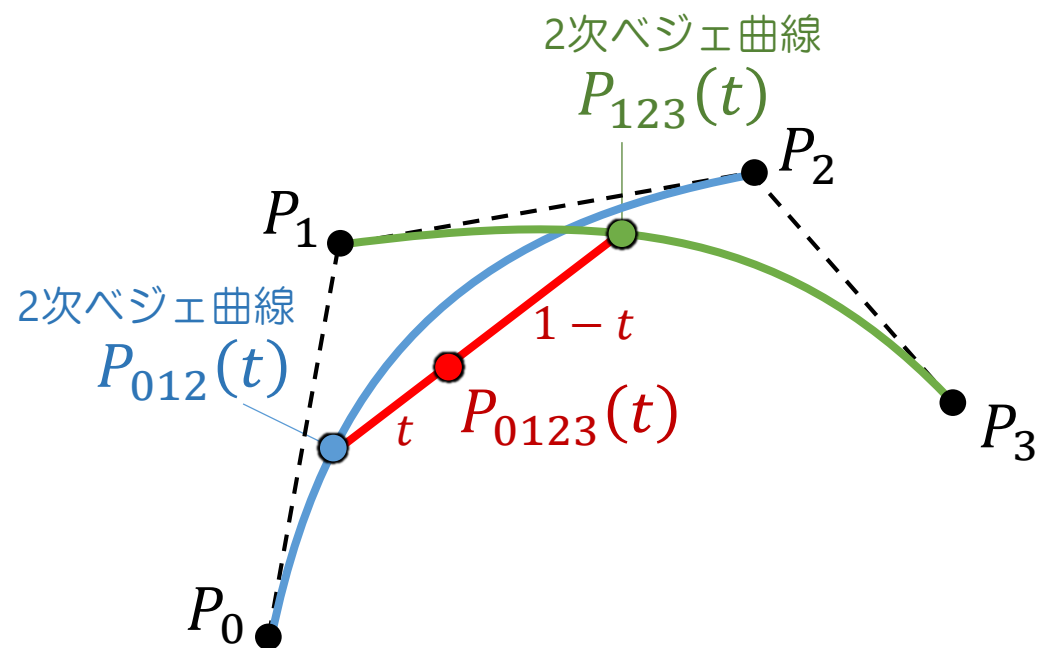
2次ベジエ曲線

3次ベジエ曲線

- 全く同じ考え方を4点 P_0, P_1, P_2, P_3 に適用：

$t: 0 \rightarrow 1$ のとき $P_{012} \rightarrow P_{123}$

となるように線形補間



- $P_{0123}(t) = (1-t)P_{012}(t) + tP_{123}(t)$

$$= (1-t)\{(1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2\} + t\{(1-t)^2P_1 + 2t(1-t)P_2 + t^2P_3\}$$

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

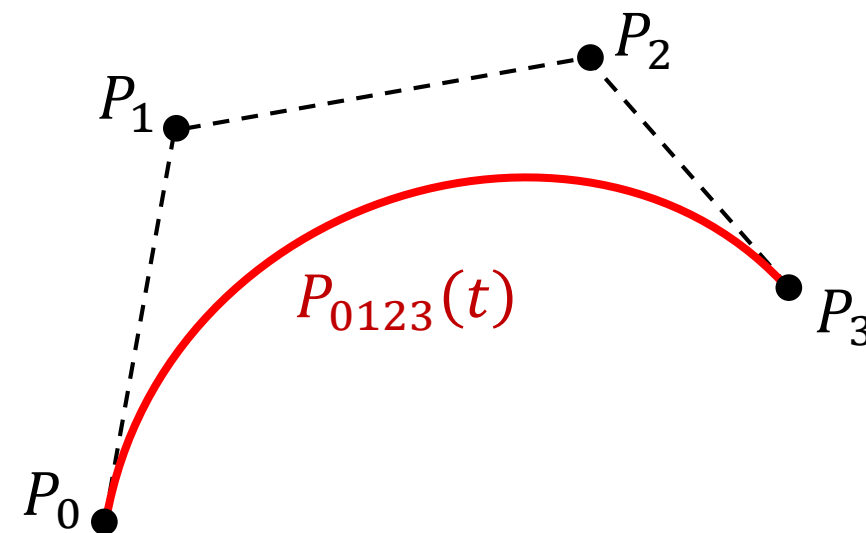
3次ベジエ曲線

3次ベジエ曲線

- 全く同じ考え方を4点 P_0, P_1, P_2, P_3 に適用：

$t: 0 \rightarrow 1$ のとき $P_{012} \rightarrow P_{123}$

となるように線形補間



- $P_{0123}(t) = (1-t)P_{012}(t) + tP_{123}(t)$

$$= (1-t)\{(1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2\} + t\{(1-t)^2P_1 + 2t(1-t)P_2 + t^2P_3\}$$

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

3次ベジエ曲線

- 両端における接線の制御がしやすい→CGで頻繁に使われる

n次ベジエ曲線

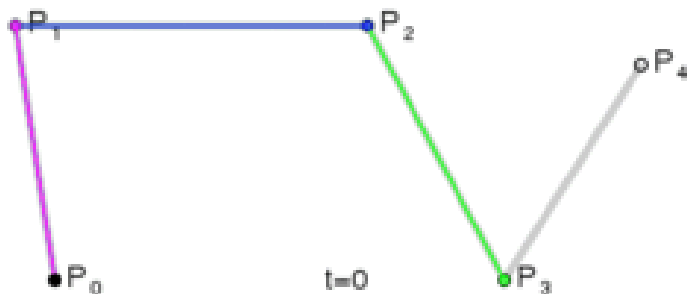
- 入力：n+1個の制御点 P_0, \dots, P_n

$$P(t) = \sum_{i=0}^n {}_n C_i t^i (1-t)^{n-i} P_i$$

$$b_i^n(t)$$

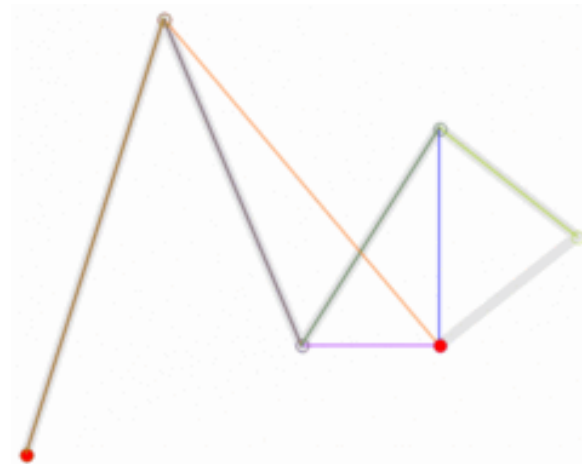
バーンスタイン基底関数

4次ベジエ曲線



$$\begin{aligned} &(1-t)^4 P_0 + \\ &4t(1-t)^3 P_1 + \\ &6t^2(1-t)^2 P_2 + \\ &4t^3(1-t) P_3 + \\ &t^4 P_4 \end{aligned}$$

5次ベジエ曲線



$$\begin{aligned} &(1-t)^5 P_0 + \\ &5t(1-t)^4 P_1 + \\ &10t^2(1-t)^3 P_2 + \\ &10t^3(1-t)^2 P_3 + \\ &5t^4(1-t) P_4 + \\ &t^5 P_5 \end{aligned}$$

3次ベジエ曲線と3次エルミート曲線の関係

- 3次ベジエ曲線とその微分：

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

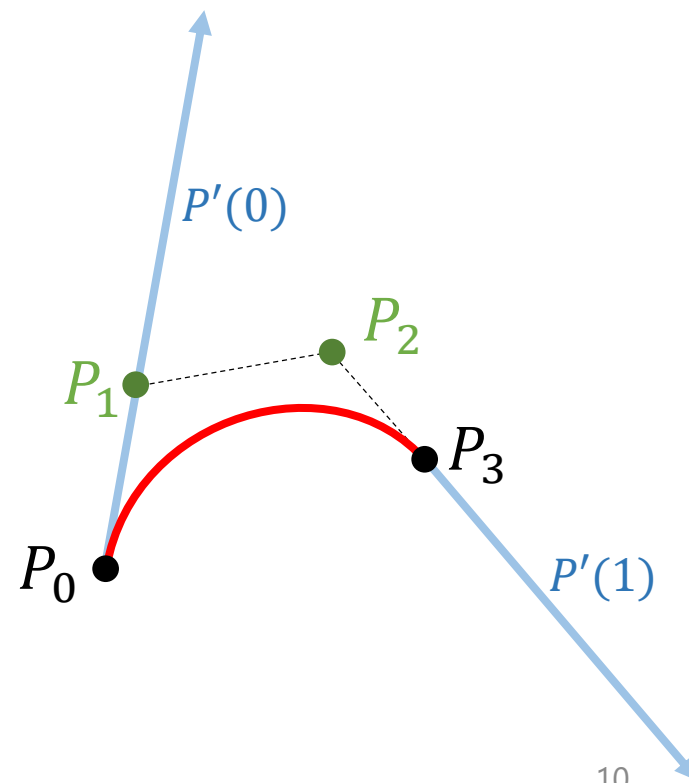
- $P'(t) = -3(1-t)^2 P_0 + 3\{(1-t)^2 - 2t(1-t)\} P_1 + 3\{2t(1-t) - t^2\} P_2 + 3t^2 P_3$

- 端点での微分値：


- $P'(0) = -3P_0 + 3P_1 \quad \rightarrow \quad P_1 = P_0 + \frac{1}{3} P'(0)$

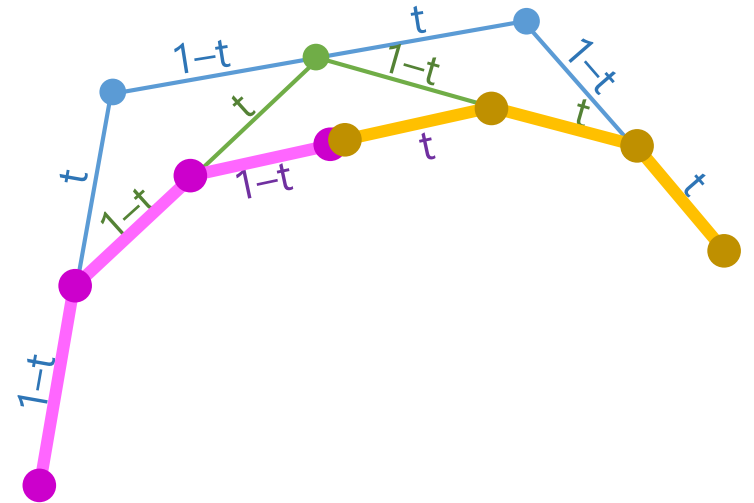
- $P'(1) = -3P_2 + 3P_3 \quad \rightarrow \quad P_2 = P_3 - \frac{1}{3} P'(1)$

- 3次多項式曲線を定義する形式が違っただけで、本質的には同じもの



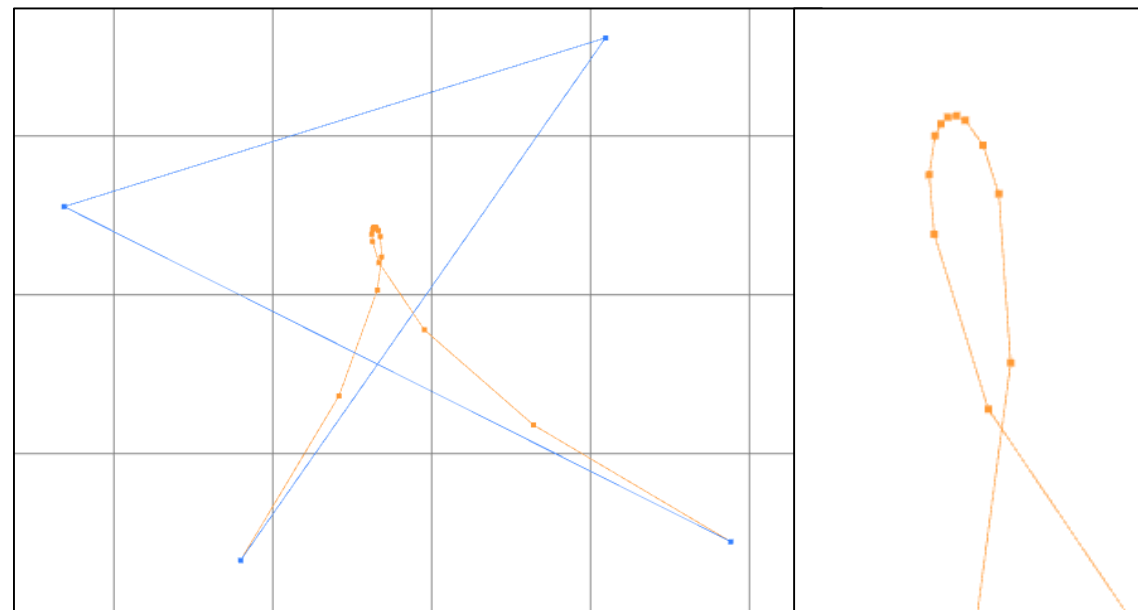
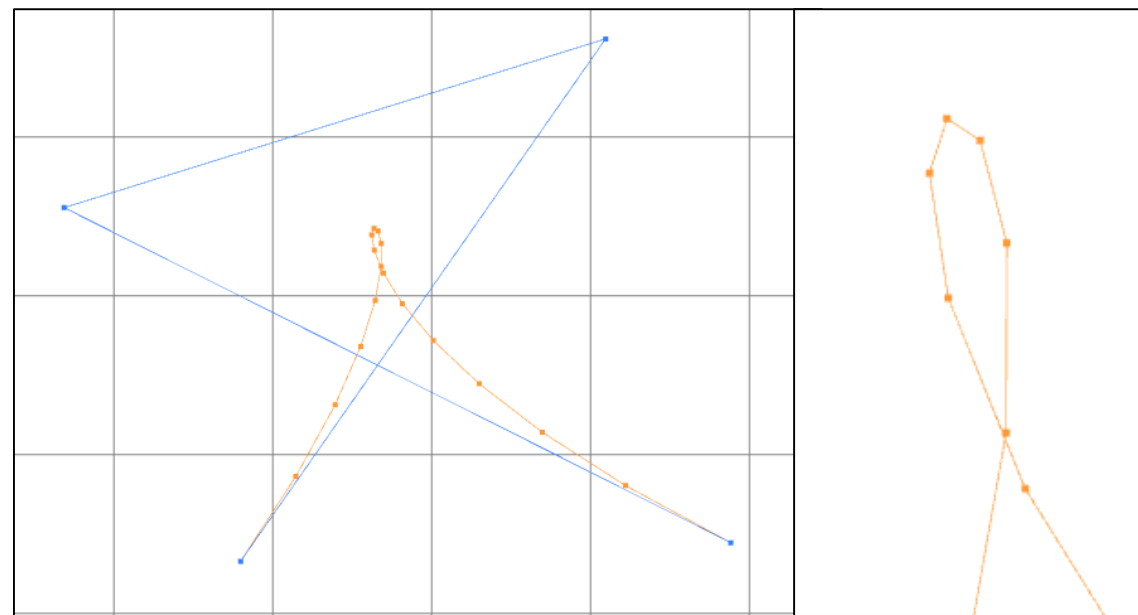
ベジエ曲線の評価方法

- 方法1：多項式をそのまま評価する
 - 単純で速いが、数値的に不安定になる場合も
- 方法2：ド・カステリヨ [de Casteljau - ベジエ曲線の再帰的な定義そのものに倣う
- 計算手順は増えるが、数値的に安定
- ベジエ曲線の分割にも使える



ベジエ曲線の描画方法

- 最終的には折れ線で近似的に描く
 - パラメタ t をどうサンプリングするかが問題
- 方法1：一定間隔でサンプリング
 - 実装が簡単
 - サンプル点の密度が必要十分でなくなる？
- 方法2：適応的なサンプリング
 - 制御点列が直線状でなければde Casteljauの方法で分割する



さらなる制御法：有理ベジエ曲線

- ベジエ曲線は、制御点の“重み付き平均”と見ることができる

- $$P_{012}(t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2$$
$$= \lambda_0(t) P_0 + \lambda_1(t) P_1 + \lambda_2(t) P_2$$

- 重要な性質：**partition of unity**

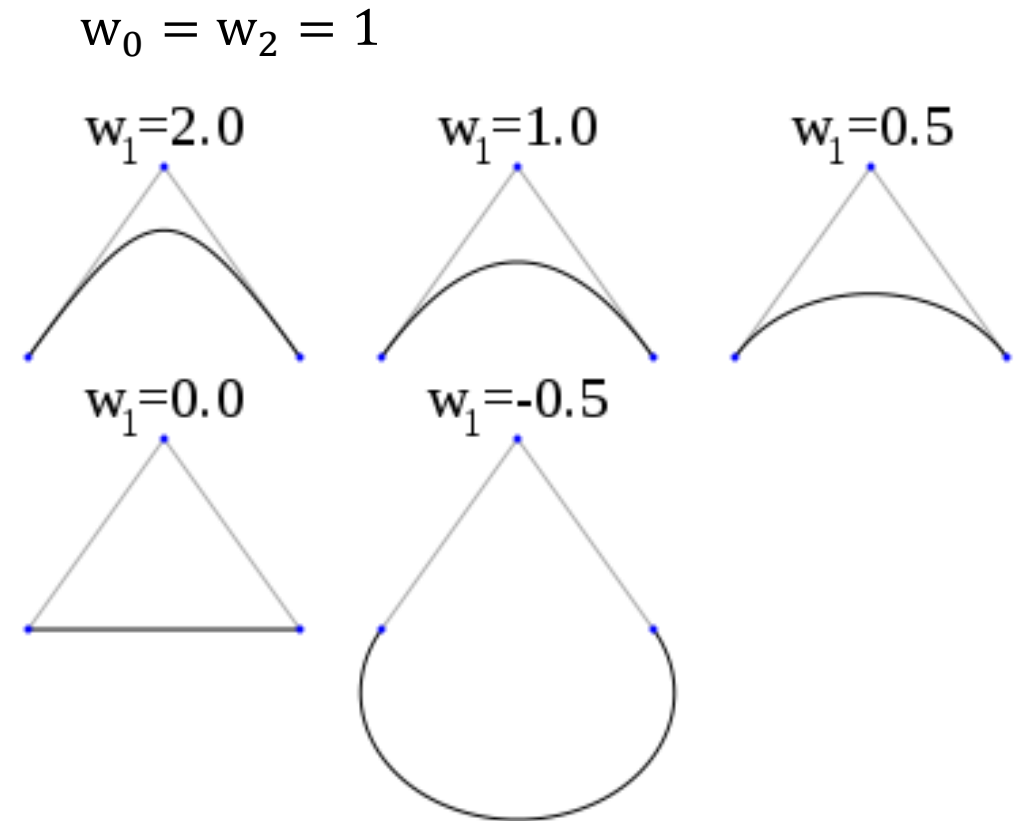
$$\lambda_0(t) + \lambda_1(t) + \lambda_2(t) = 1 \quad \forall t$$

- 各制御点の重み $\lambda_i(t)$ に任意係数 w_i を掛ける：

$$\xi_i(t) = w_i \lambda_i(t)$$

- 正規化して新しい重みを得る：

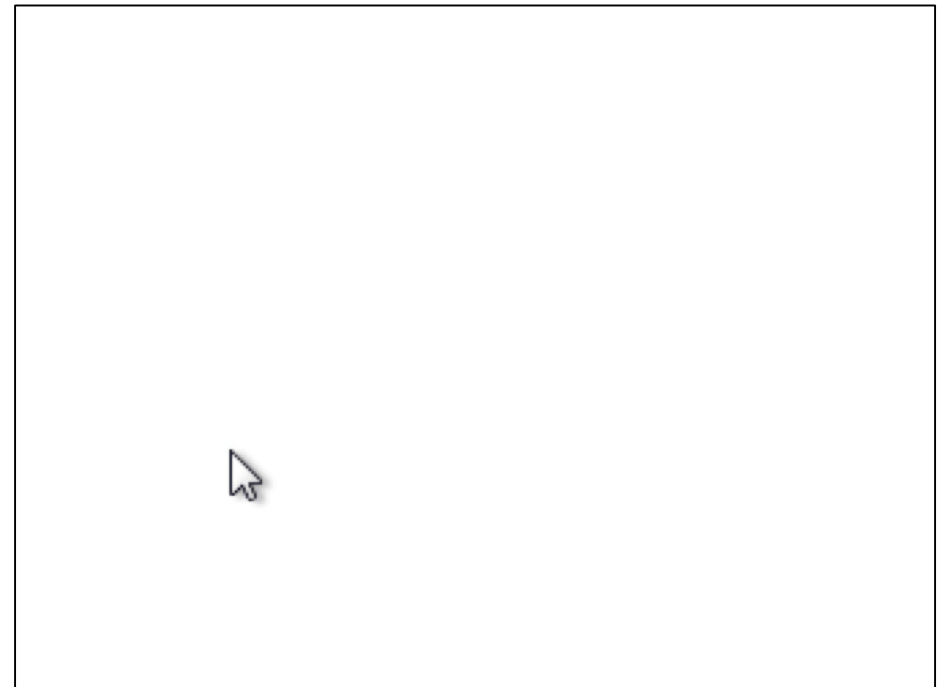
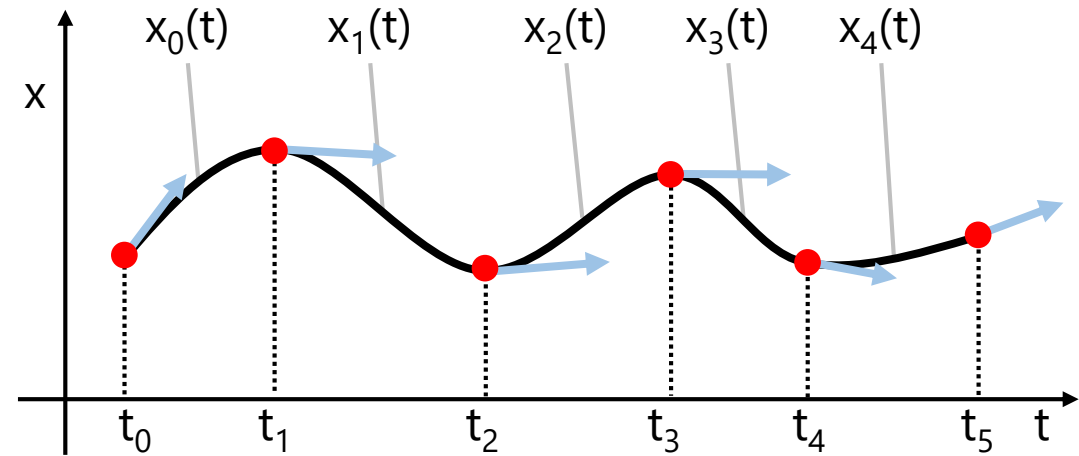
$$\lambda'_i(t) = \frac{\xi_i(t)}{\sum_j \xi_j(t)}$$



多項式曲線ではない → 円弧なども表現可能

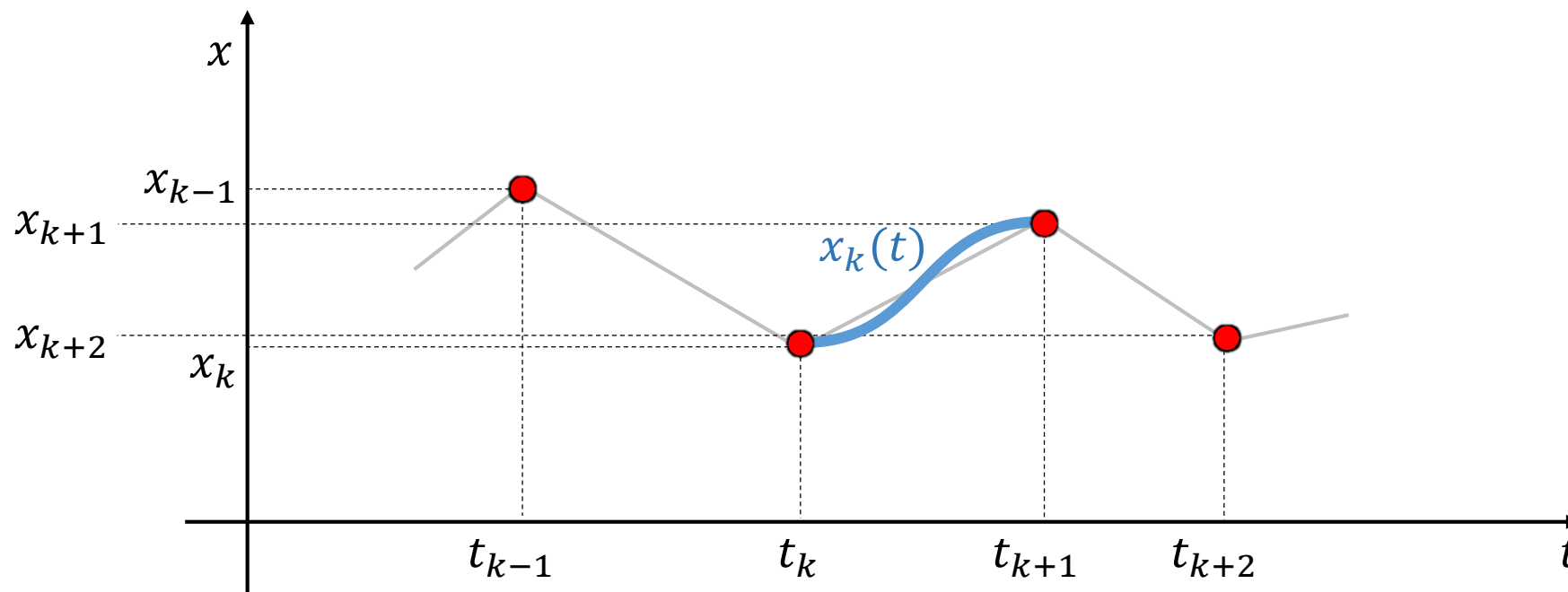
3次スプライン

- 複数の3次曲線を滑らかに繋げたもの
 - 区分的多項式
 - 区分の境目で値と微分値が共通 (C^1 連続)
- パラメタ範囲は $[0, 1]$ とは限らない
 - ただし $t_k < t_{k+1}$
- 値のみを入力として、
微分値を適切に自動推定したい



3次Catmull-Romスプライン

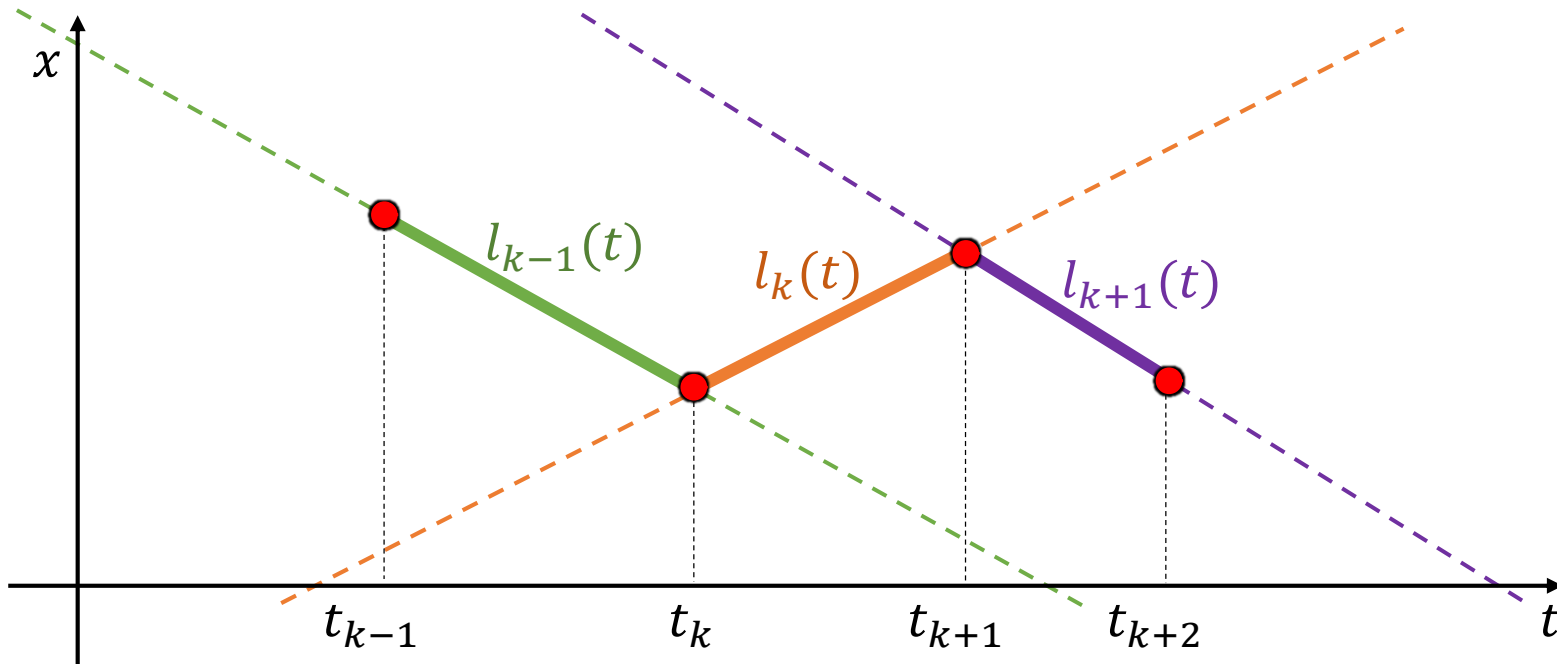
- 区間 $t_k \leq t \leq t_{k+1}$ における3次関数 $x_k(t)$ を、前後の制約値 $x_{k-1}, x_k, x_{k+1}, x_{k+2}$ から決定



3次Catmull-Romスプライン：ステップ1

- $t_k \rightarrow t_{k+1}$ のとき $x_k \rightarrow x_{k+1}$ となるように補間 \rightarrow 直線

$$l_k(t) = \left(1 - \frac{t - t_k}{t_{k+1} - t_k}\right) x_k + \frac{t - t_k}{t_{k+1} - t_k} x_{k+1}$$

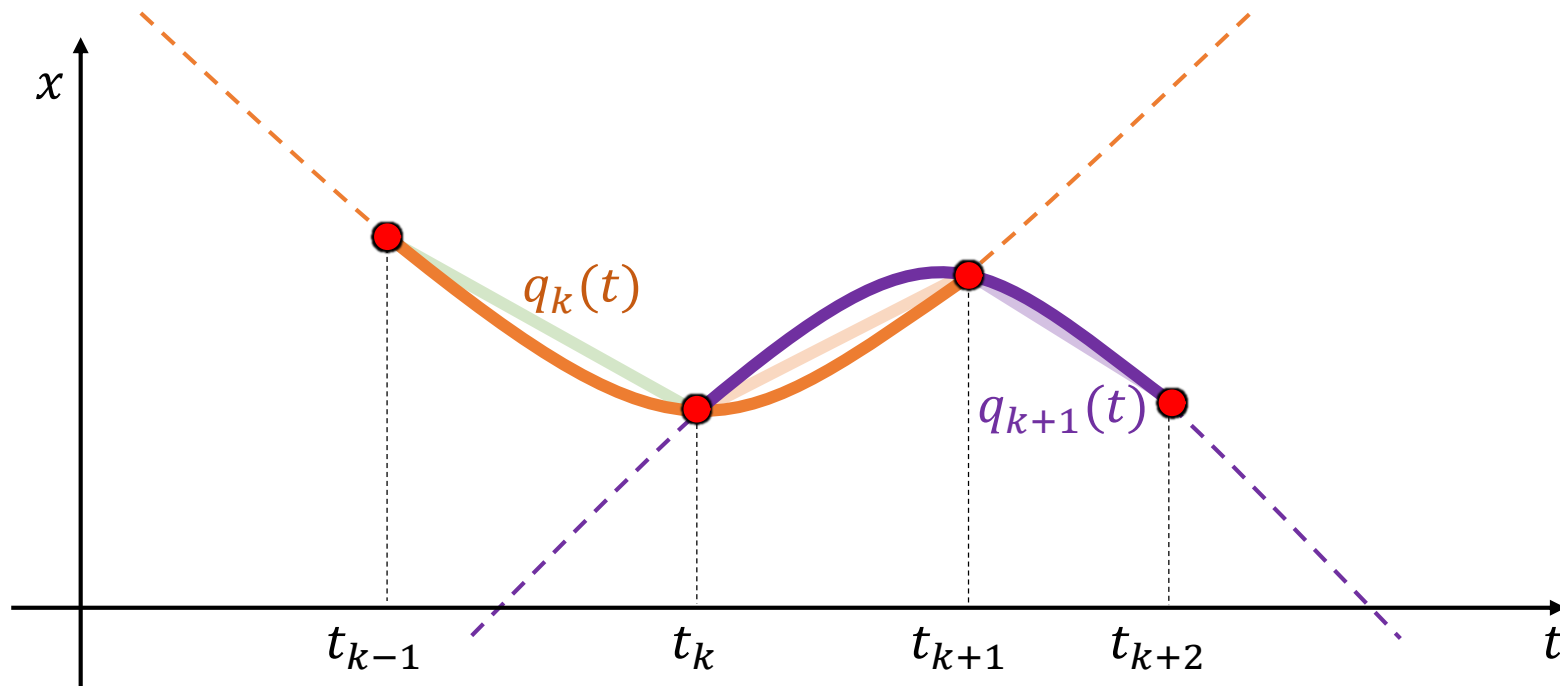


3次Catmull-Romスプライン：ステップ2

- $t_{k-1} \rightarrow t_{k+1}$ のとき $l_{k-1} \rightarrow l_k$ となるように補間 \rightarrow 2次曲線

$$q_k(t) = \left(1 - \frac{t - t_{k-1}}{t_{k+1} - t_{k-1}}\right) l_{k-1}(t) + \frac{t - t_{k-1}}{t_{k+1} - t_{k-1}} l_k(t)$$

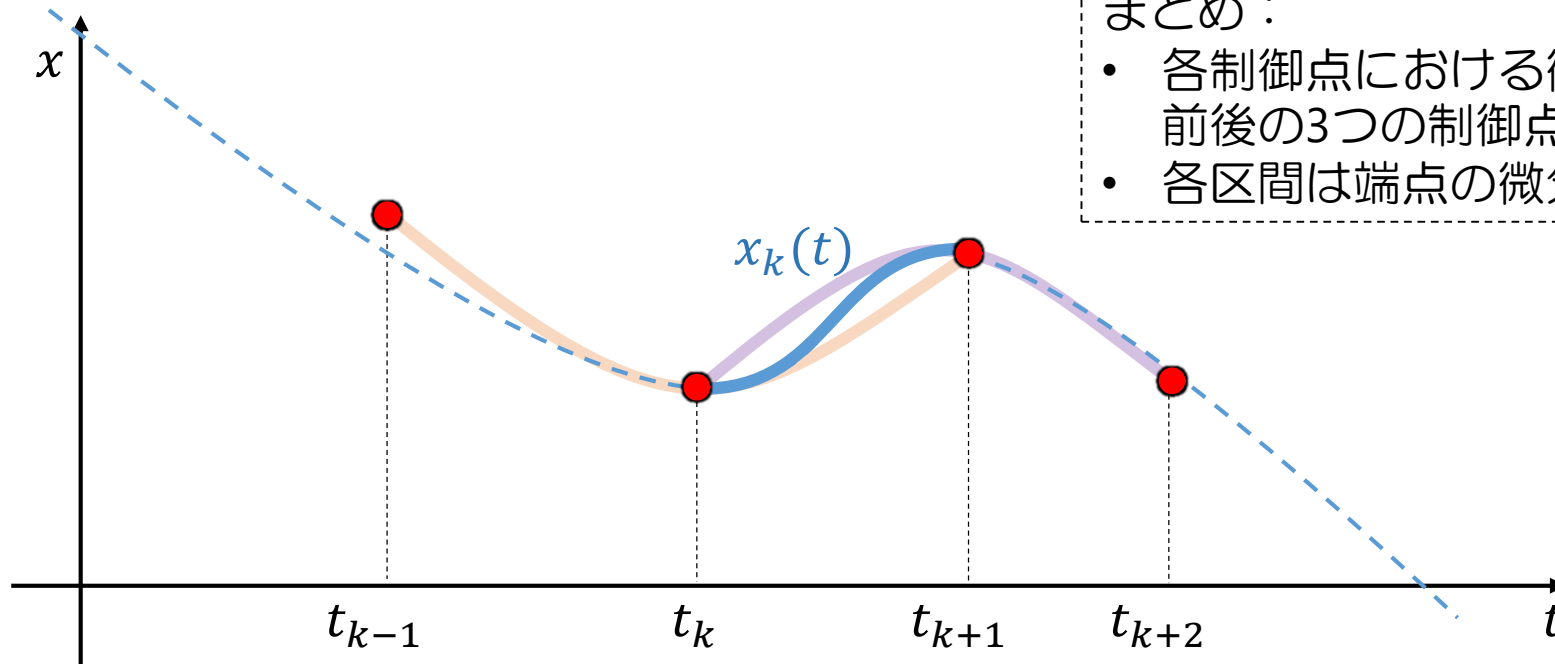
- 3点 $(t_{k-1}, x_{k-1}), (t_k, x_k), (t_{k+1}, x_{k+1})$ を通る



3次Catmull-Romスプライン：ステップ3

- $t_k \rightarrow t_{k+1}$ のとき $q_k \rightarrow q_{k+1}$ となるように補間 \rightarrow 3次曲線

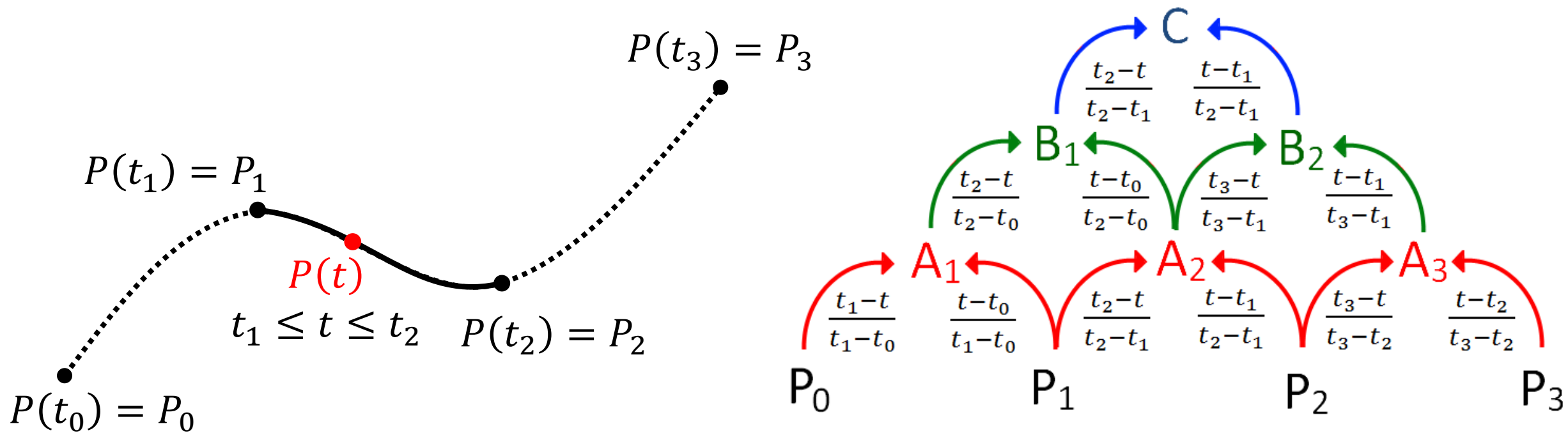
$$x_k(t) = \left(1 - \frac{t - t_k}{t_{k+1} - t_k}\right) q_k(t) + \frac{t - t_k}{t_{k+1} - t_k} q_{k+1}(t)$$



まとめ：

- 各制御点における微分値が、それ自身とその前後の3つの制御点を通る2次曲線から定まる
- 各区間は端点の微分値を満たす3次曲線

3次Catmull-Romスプラインの計算方法



パラメタ区分 t_k (ノット列) の決め方

- 前提: $t_0 = 0$

- Uniform (一様)

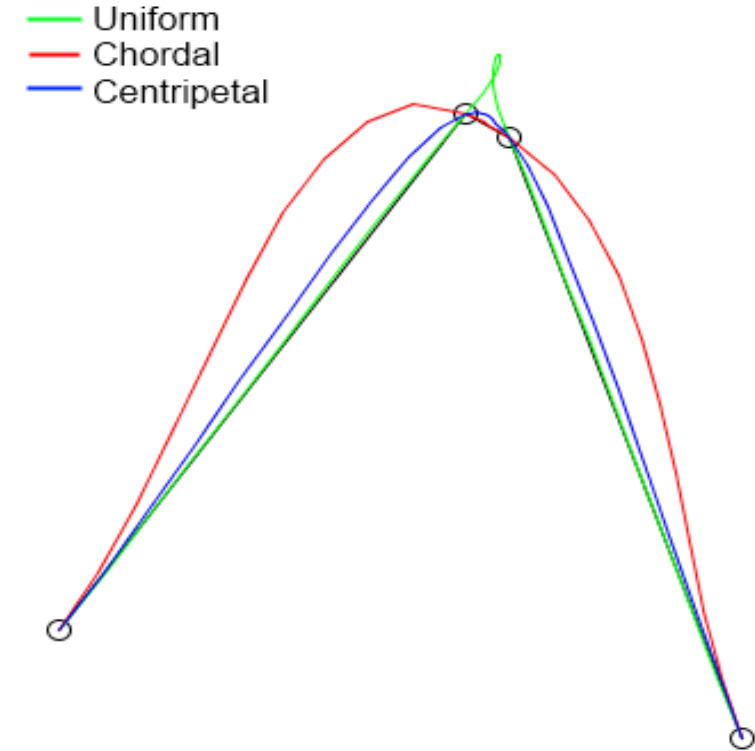
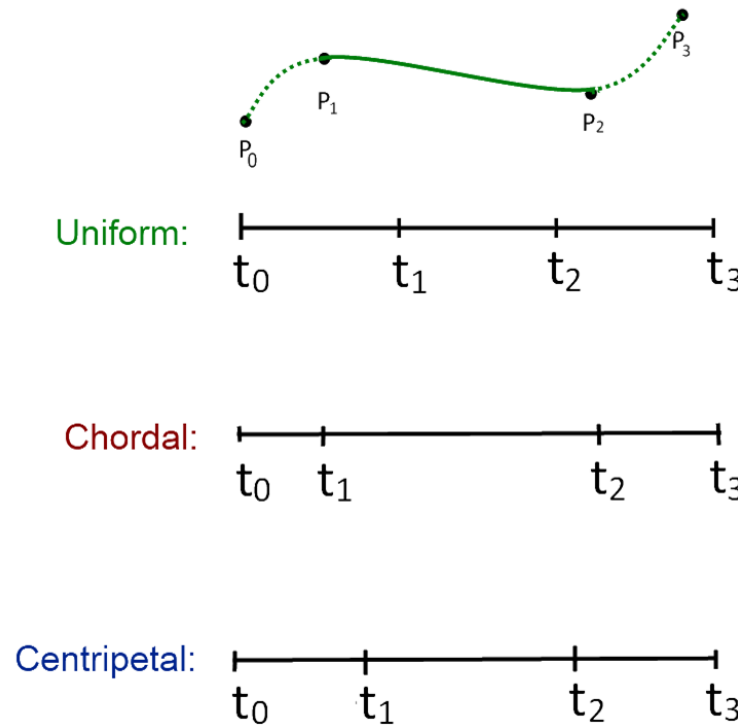
$$t_k = t_{k-1} + 1$$

- Chordal (弧長に基づく)

$$t_k = t_{k-1} + |P_{k-1} - P_k|$$

- Centripetal (求心性?)

$$t_k = t_{k-1} + \sqrt{|P_{k-1} - P_k|}$$

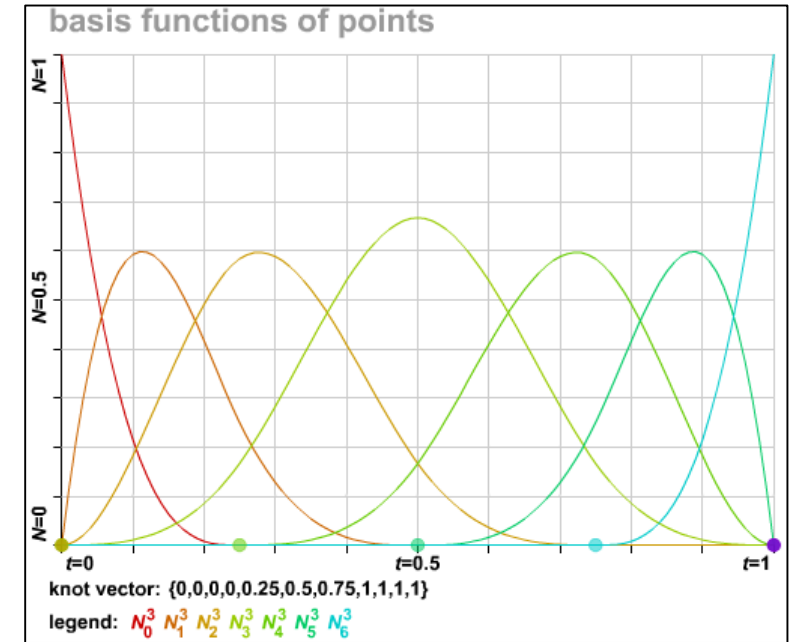


3次Catmull-Romスプラインの応用

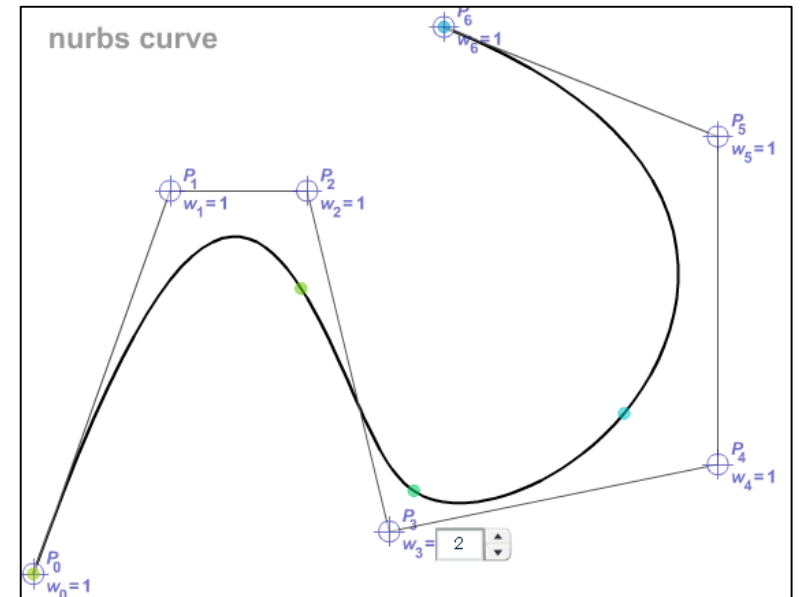


Bスプライン曲線

- 多項式スプラインを表すもう一つの方法
 - 曲線を**基底関数 (basis)** の重ね合わせで表す
 - 3次の基底関数が実用上一般的
- **サブディビジョンサーフェス**と深い関係
→ 次回講義で説明



- **Non-Uniform Rational B-Spline**
 - Non-Uniform = ノット列 (t_k) の間隔が一様でない
 - Rational = 制御点が重みを持つ (有理式曲線)
 - (ややこしいので本講義では説明しない)
- 秀逸なFlashデモ：
<http://geometrie.foretnik.net/files/NURBS-en.swf>

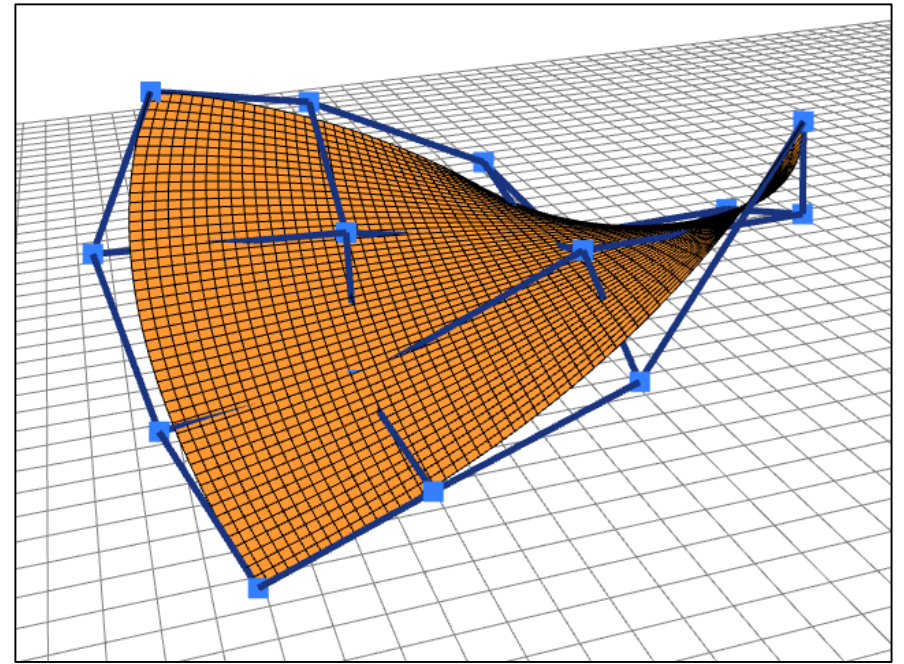


パラメトリック曲面

- パラメタが1個 → 曲線 $P(t)$
- パラメタが2個 → 曲面 $P(s, t)$

- 3次ベジエ曲面：
 - 入力： $4 \times 4 = 16$ 個の制御点 P_{ij}

$$P(s, t) = \sum_{i=0}^3 \sum_{j=0}^3 b_i^3(s) b_j^3(t) P_{ij}$$



バーンスタイン基底関数

$$b_0^3(t) = (1 - t)^3$$

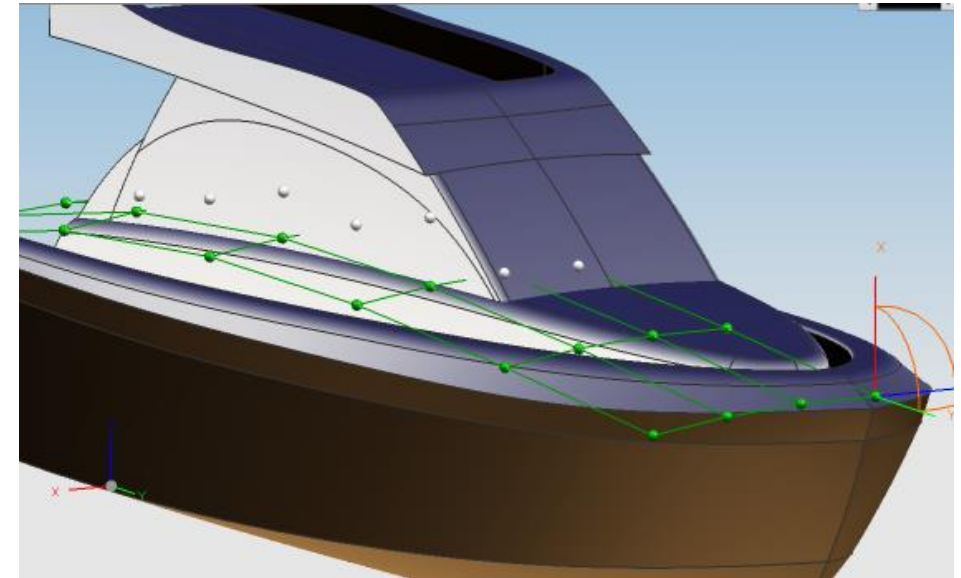
$$b_1^3(t) = 3t(1 - t)^2$$

$$b_2^3(t) = 3t^2(1 - t)$$

$$b_3^3(t) = t^3$$

パラメトリック曲面を用いた3Dモデリング

- 長所
 - 滑らかな曲面をコンパクトに表現できる
 - 球や円錐面などを正確に表現できる
- 短所
 - 複数のパッチをうまく配置するのが難しい
 - 複数のパッチ間の連続性を保つのが難しい
- シンプルなパーツの組み合わせで人工物をデザインするのによく使われる



参考

- http://en.wikipedia.org/wiki/Bezier_curve
- http://antigrain.com/research/adaptive_bezier/
- <https://groups.google.com/forum/#!topic/comp.graphics.algorithms/2FypAv29dG4>
- http://en.wikipedia.org/wiki/Cubic_Hermite_spline
- http://en.wikipedia.org/wiki/Centripetal_Catmull%E2%80%93Rom_spline