コンピュータグラフィクス論

– モデリング (1) –

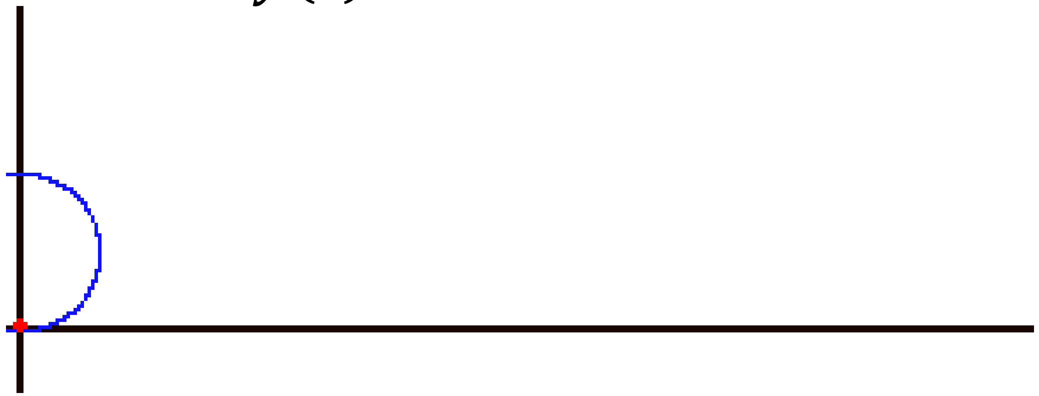
2015年4月16日

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 によって決まるもの
  - 例：サイクロイド

$$x(t) = t - \sin t$$
$$y(t) = 1 - \cos t$$

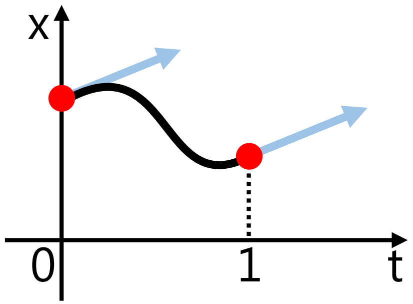


- 接線ベクトル：$\left(x'(t), y'(t)\right)$
- 多項式曲線： $x(t) = \sum_i a_i t^i$

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

$$x(0) = x_0$$
$$x(1) = x_1$$
$$x'(0) = x'_0$$
$$x'(1) = x'_1$$



- 制約が4つなので、4自由度が必要
  ➔3次多項式
  - $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$
  - $x'(t) = a_1 + 2a_2 t + 3a_3 t^2$

$$x(0) = a_0 = x_0$$
$$x(1) = a_0 + a_1 + a_2 + a_3 = x_1$$
$$x'(0) = a_1 = x'_0$$
$$x'(1) = a_1 + 2\,a_2 + 3\,a_3 = x'_1$$

➔

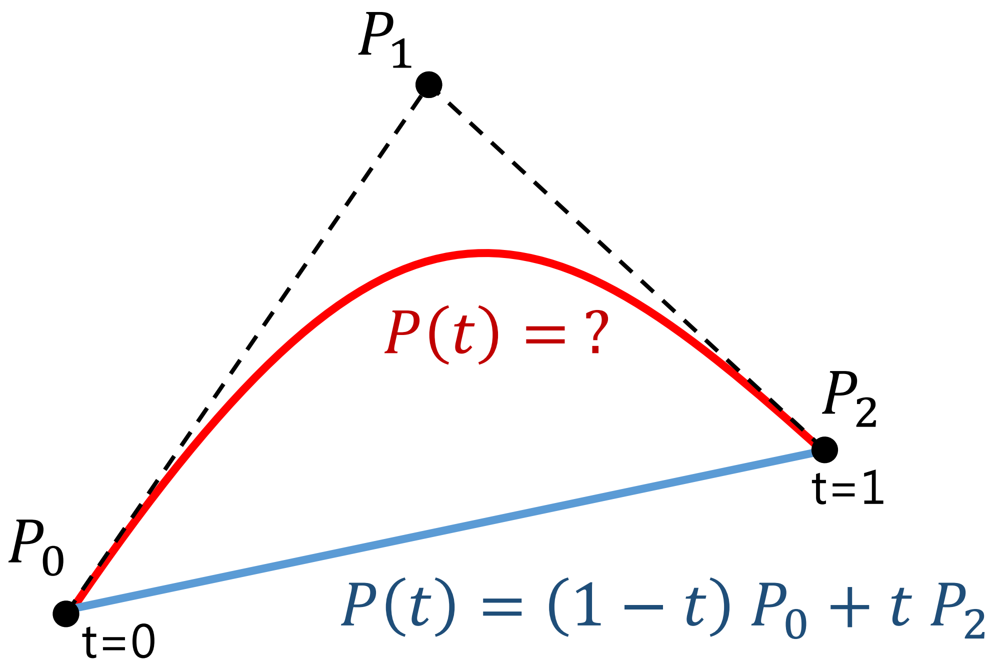
$$a_0 = x_0$$
$$a_1 = x'_0$$
$$a_2 = -3\,x_0 + 3\,x_1 - 2\,x'_0 - x'_1$$
$$a_3 = 2\,x_0 - 2\,x_1 + x'_0 + x'_1$$

- 制約式を代入すれば係数が求まる

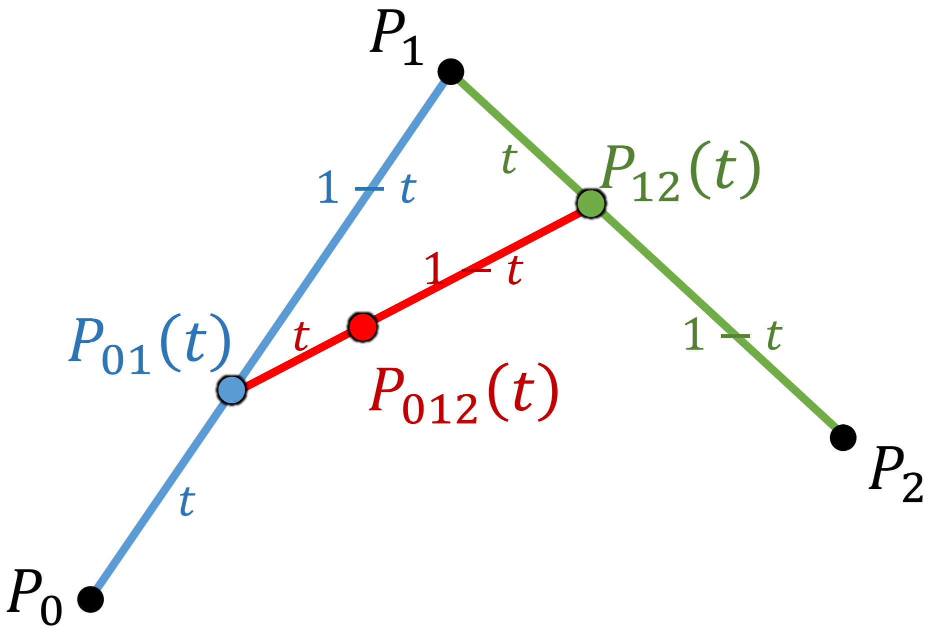
ベジェ (Bezier[ベズィエ]) 曲線

- 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



ベジェ曲線



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

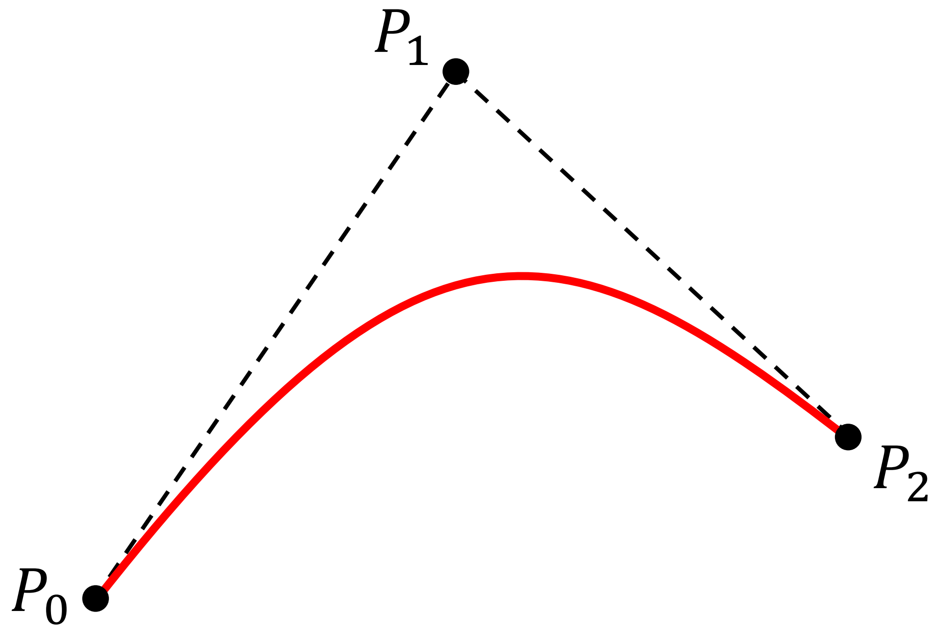
- アイディア： "補間を補間"
  $t : 0 \to 1$ のとき $P_{01} \to P_{12}$ となるように線形補間

- $P_{012}(t) = (1-t)P_{01}(t) + t\,P_{12}(t)$

$$= (1-t)\{(1-t)P_0 + t\,P_1\} + t\,\{(1-t)P_1 + t\,P_2\}$$

$$= \underbrace{(1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2}_{\text{2次ベジェ曲線}}$$

ベジェ曲線



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

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

- $P_{012}(t) = (1-t)P_{01}(t) + t\,P_{12}(t)$

$$= (1-t)\{(1-t)P_0 + t\,P_1\} + t\,\{(1-t)P_1 + t\,P_2\}$$

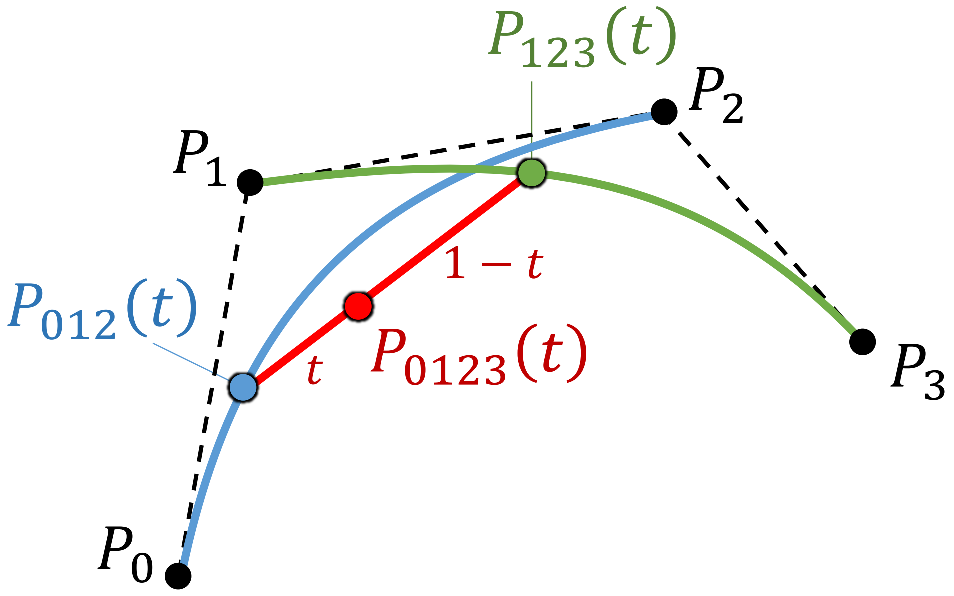
$$= \underbrace{(1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2}_{\text{2次ベジェ曲線}}$$

3次ベジェ曲線

- 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



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

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

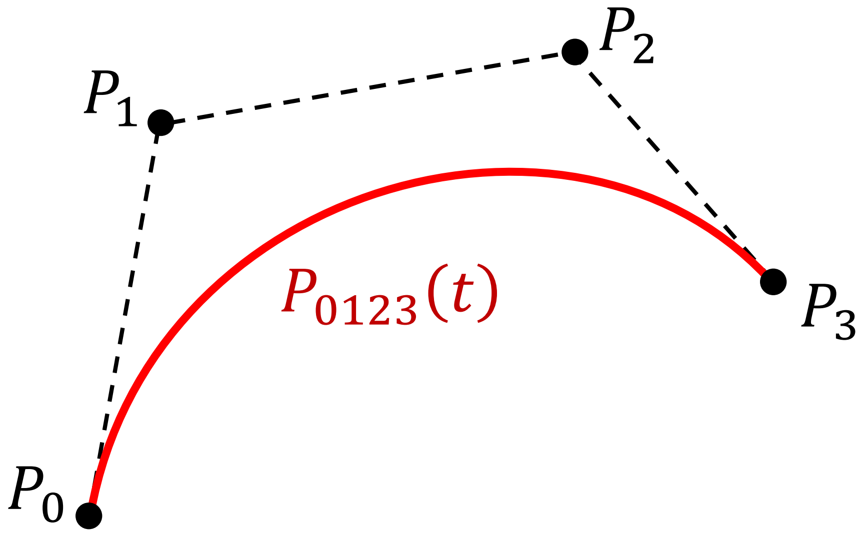
$$= \underbrace{(1-t)^3P_0 + 3t(1-t)^2P_1 + 3t^2(1-t)P_2 + t^3P_3}_{\text{3次ベジェ曲線}}$$

3次ベジェ曲線

- 全く同じ考え方を4点P_0, P_1, P_2 P_3に適用：

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

  となるように線形補間



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

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

$$= \underbrace{(1-t)^3P_0 + 3t(1-t)^2P_1 + 3t^2(1-t)P_2 + t^3P_3}_{\text{3次ベジェ曲線}}$$

- 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

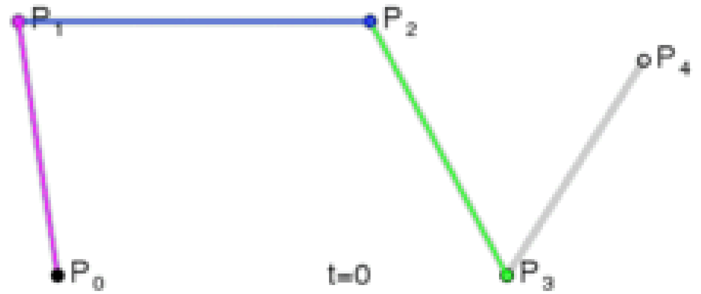
n次ベジェ曲線

- 入力：n+1個の制御点$P_0, \cdots, P_n$

$$P(t) = \sum_{i=0}^{n} \underbrace{{}_n\mathrm{C}_i \, t^i \, (1-t)^{n-i}}_{b_i^n(t)} P_i$$

バーンスタイン基底関数

4次ベジェ曲線



$(1-t)^4 P_0 \quad +$

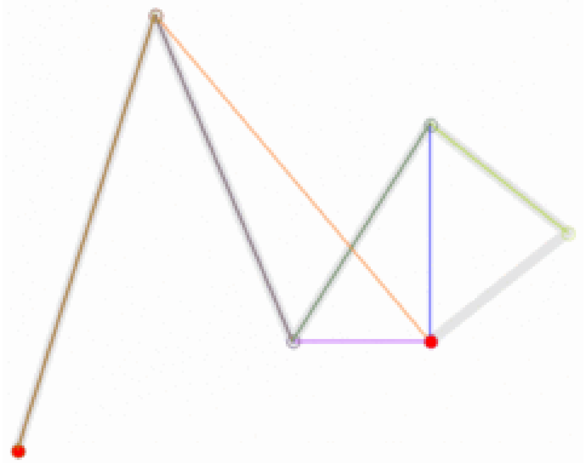
$4t(1-t)^3 P_1 \quad +$

$6t^2(1-t)^2 P_2 \quad +$

$4t^3(1-t) P_3 \quad +$

$t^4 P_4$

5次ベジェ曲線



$(1-t)^5 P_0 \quad +$

$5t(1-t)^4 P_1 \quad +$

$10t^2(1-t)^3 P_2 \quad +$

$10t^3(1-t)^2 P_3 \quad +$

$5t^4(1-t) P_4 \quad +$

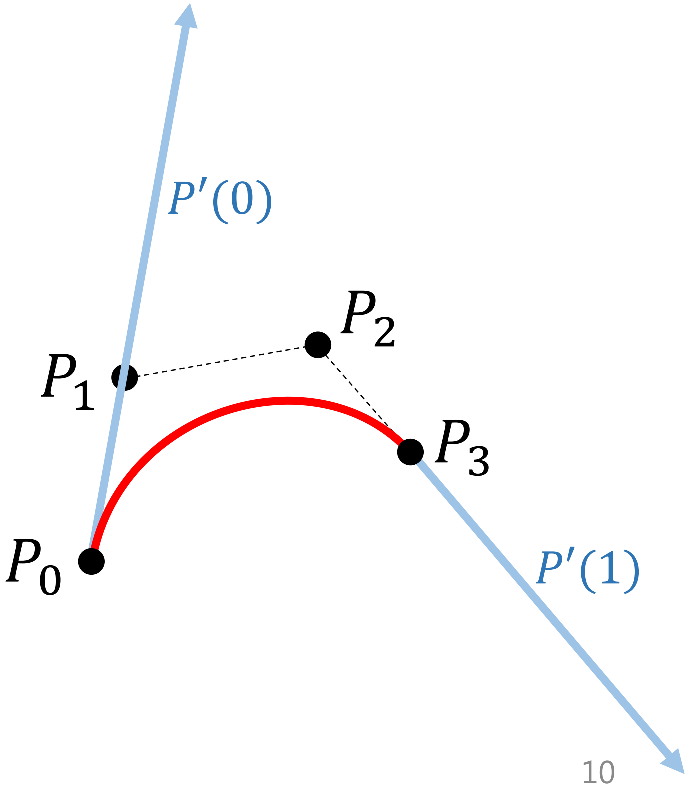
$t^5 P_5$

3次ベジェ曲線と3次エルミート曲線の関係

- 3次ベジェ曲線とその微分：
  - $P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3$
  - $P'(t) = -3(1-t)^2 P_0 + 3\{(1-t)^2 - 2t(1-t)\} P_1 + 3\{2t(1-t) - t^2\} P_2 + 3t^2 P_3$

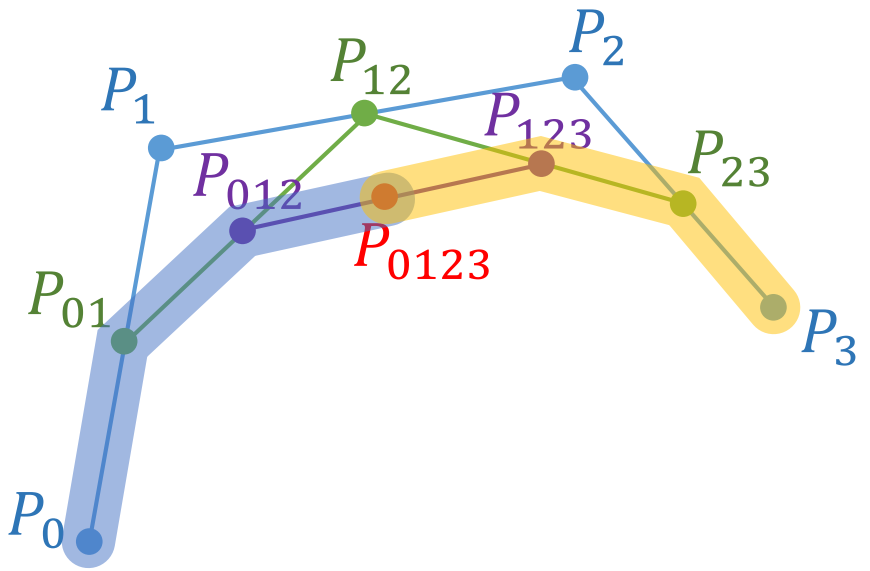
- 端点での微分値：
  - $P'(0) = -3P_0 + 3P_1$ ➔ $P_1 = P_0 + \frac{1}{3} P'(0)$
  - $P'(1) = -3P_2 + 3P_3$ ➔ $P_2 = P_3 - \frac{1}{3} P'(1)$

- 3次多項式曲線を定義する形式が違うだけ



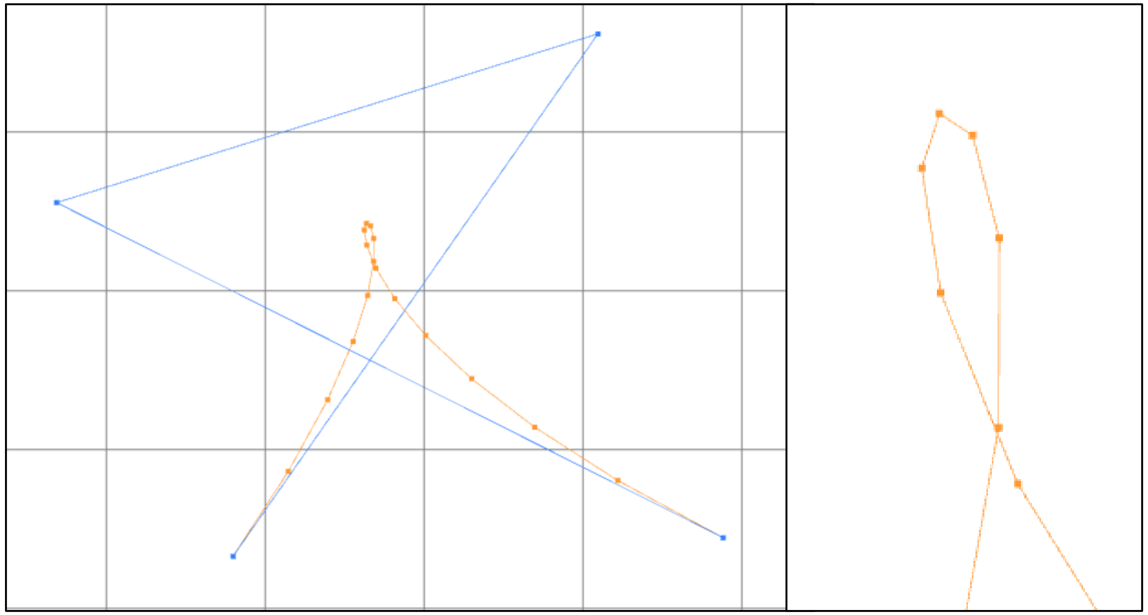
ベジェ曲線の評価方法

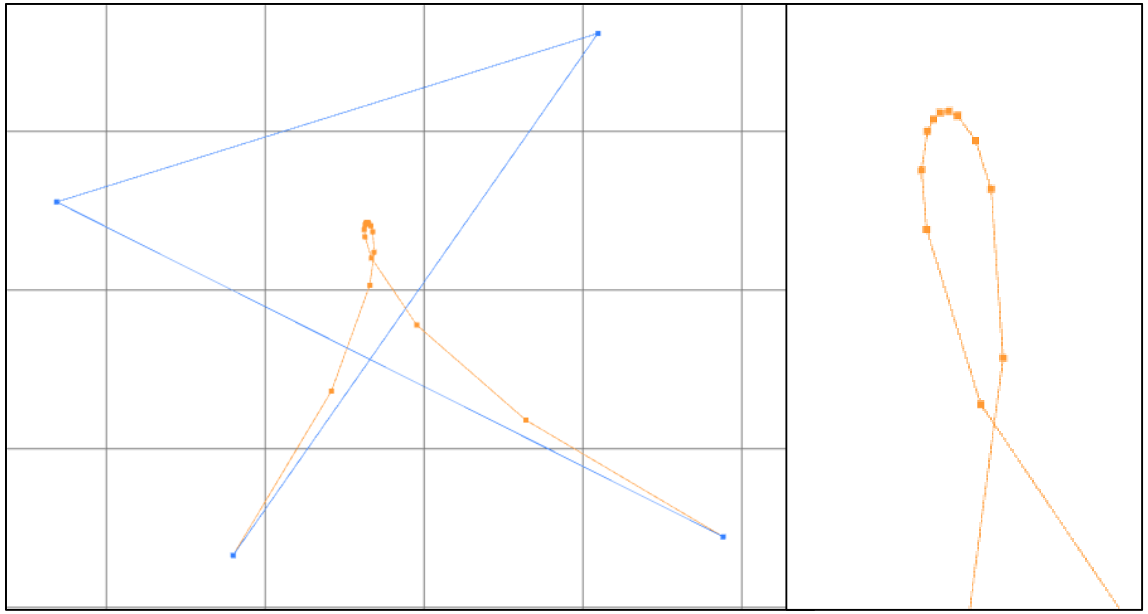
- 方法1：多項式をそのまま評価する
  - 単純で速いが、数値的に不安定になる場合も
- 方法2：ド・カステリョ [de Casteljau] のアルゴリズム
  - ベジェ曲線の再帰的な定義そのものに倣う
  - 計算手順は増えるが、数値的に安定
  - ベジェ曲線の分割にも使える



ベジェ曲線の描画方法

- 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：有理ベジェ曲線

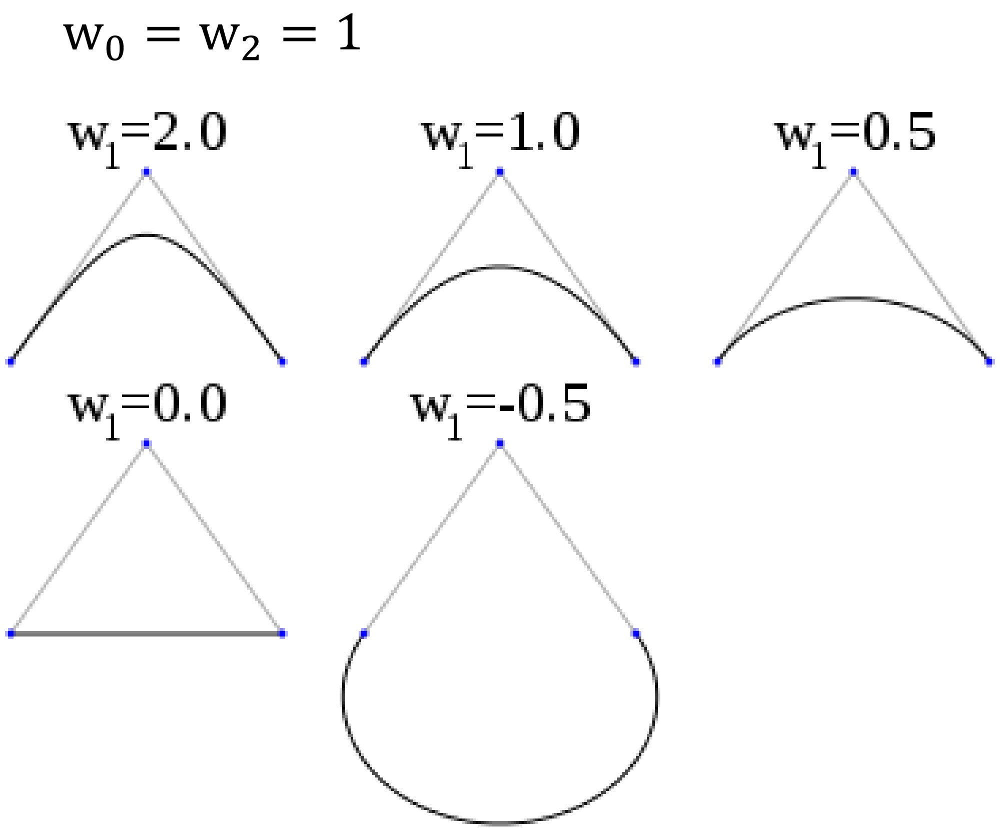
- ベジェ曲線は、制御点の “重み付き平均” と見ることができる
  - $P_{012}(t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2$
    $= \lambda_0(t) P_0 + \lambda_1(t) P_1 + \lambda_2(t) P_2$
  - 重要な性質：$\lambda_0(t) + \lambda_1(t) + \lambda_2(t) = 1 \quad \forall t$

- 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：

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



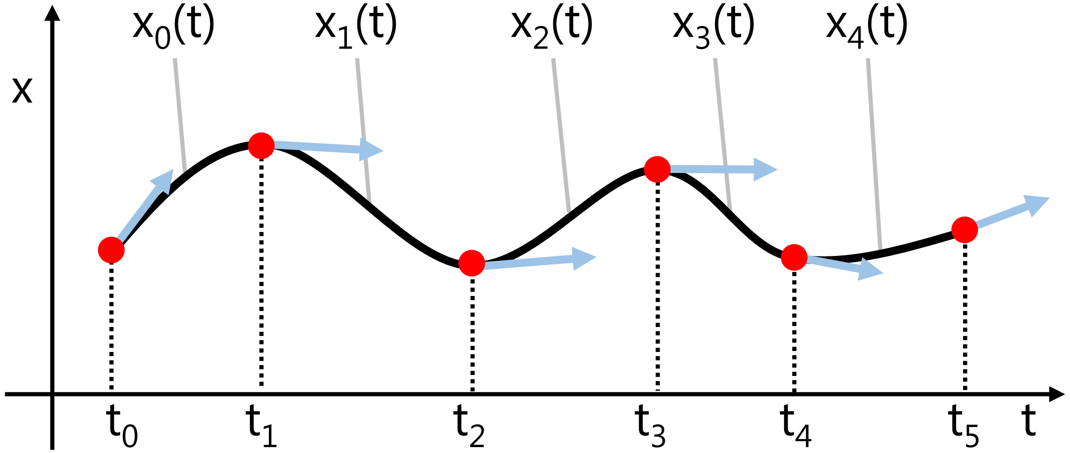
多項式曲線ではない ➔ 円弧なども表現可能

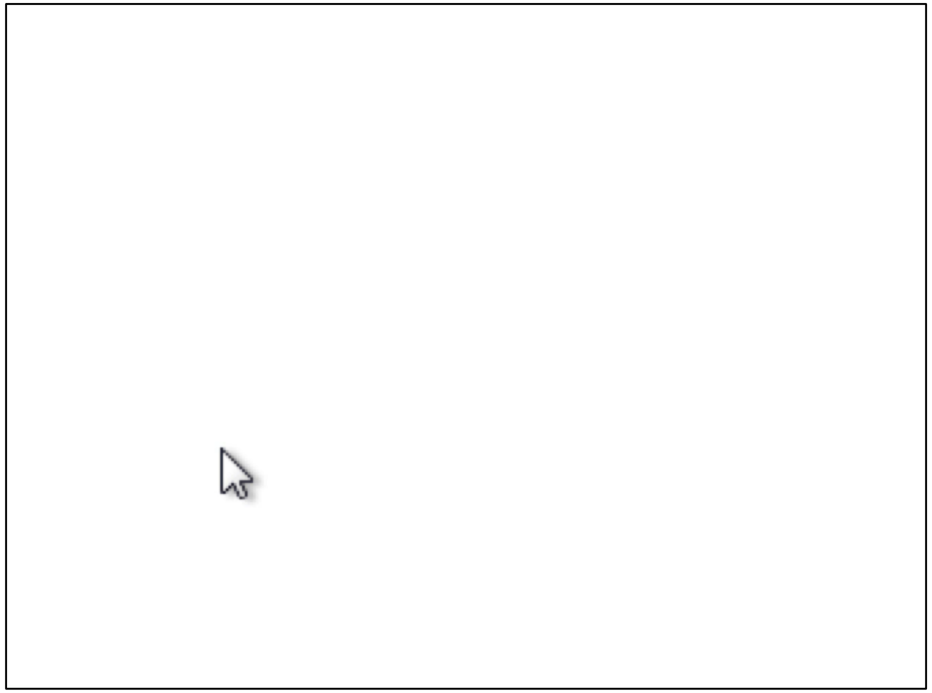
3次スプライン

- 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

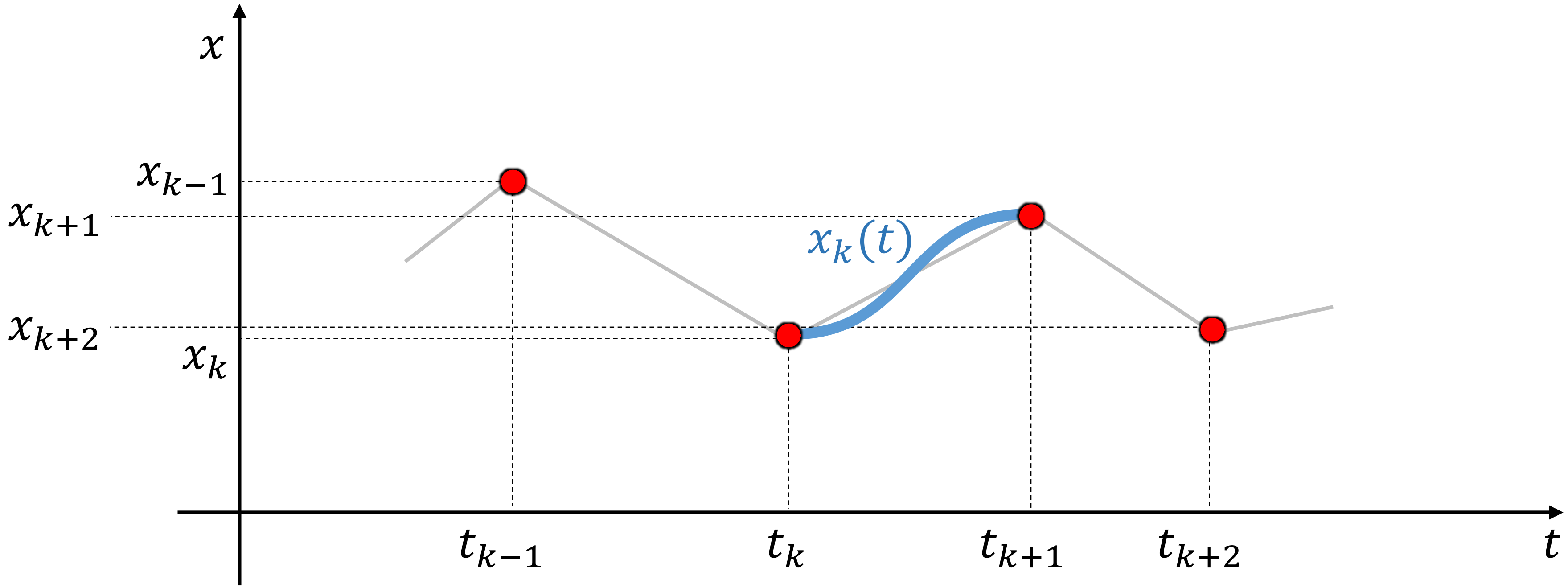




PowerPointの曲線ツール

3次Catmull-Romスプライン

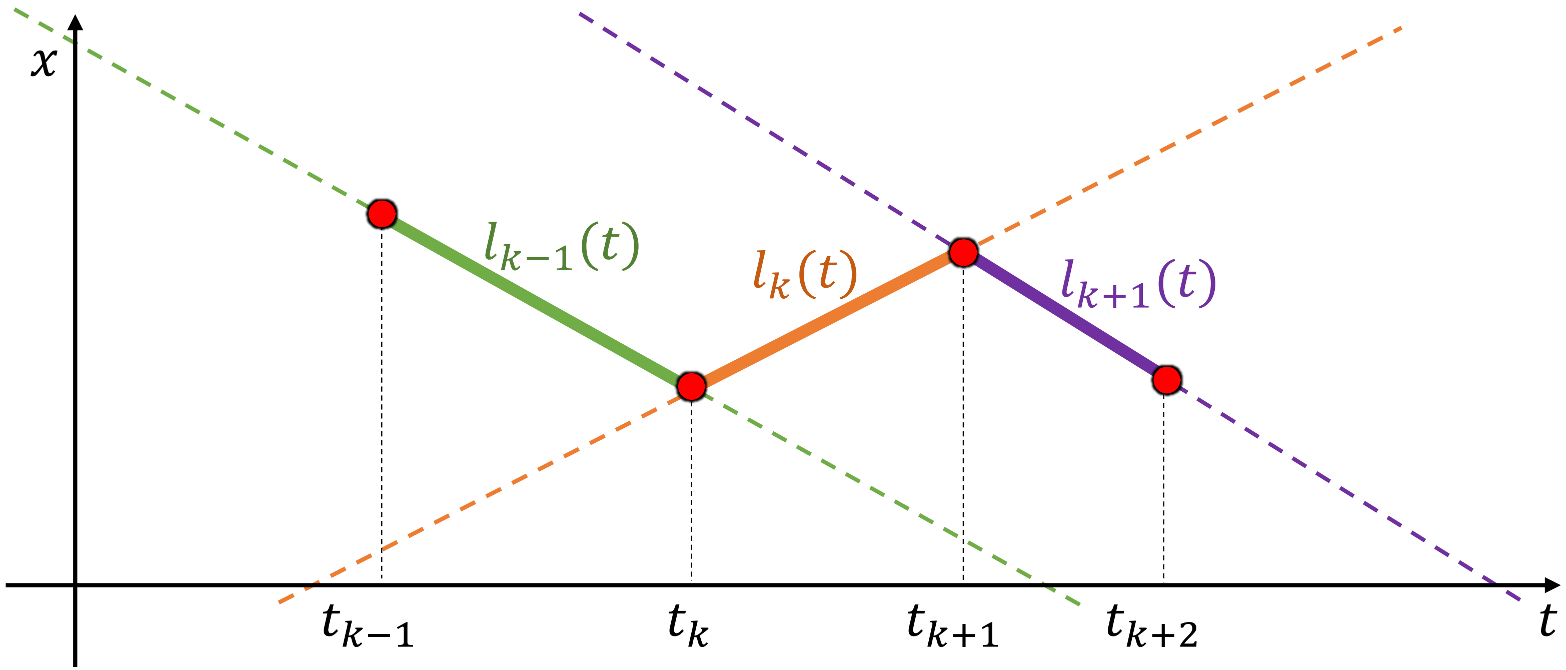
- 区間 $t_k \le t \le t_{k+1}$ における3次関数 $x_k(t)$ を、前後の制約値 $x_{k-1}, x_k, x_{k+1}, x_{k+2}$ から決定



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 ➔ 直線

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

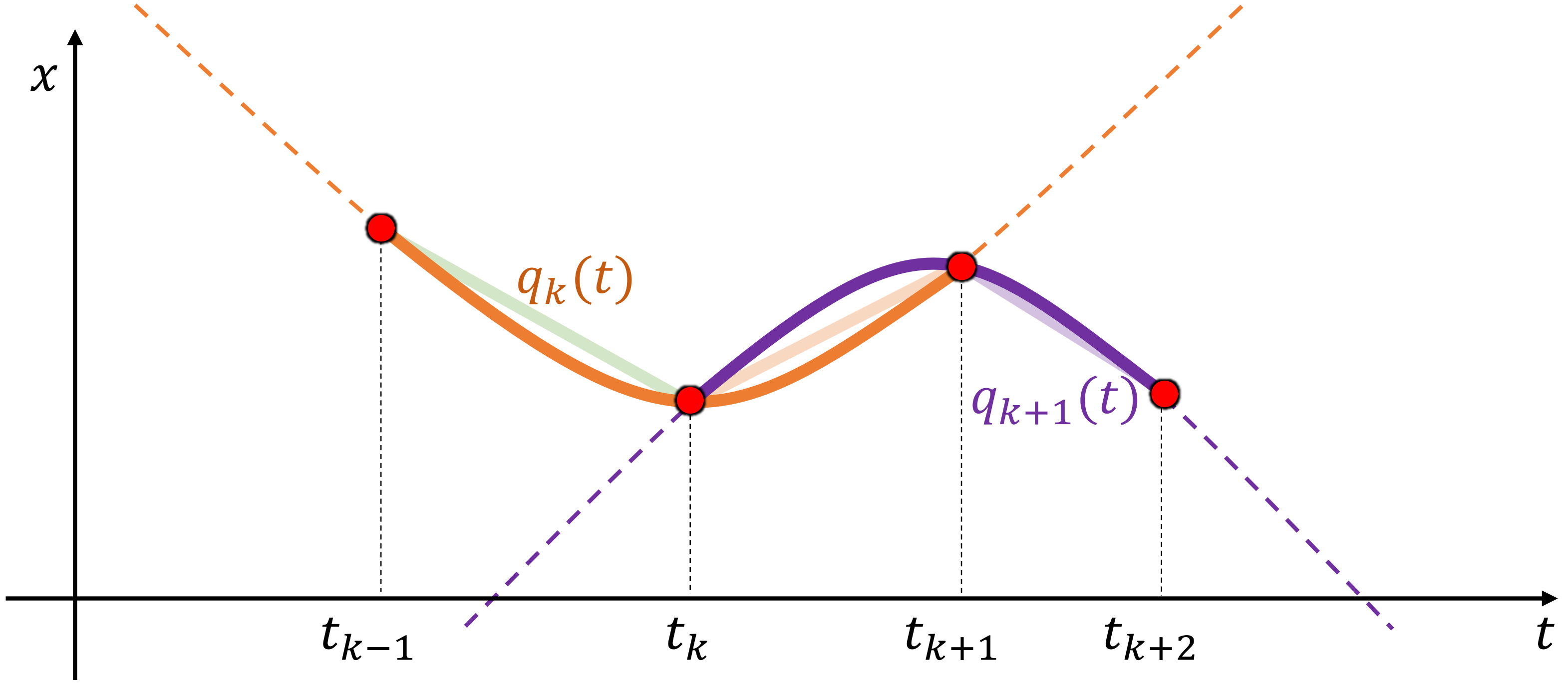


3次Catmull-Romスプライン：ステップ2

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

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

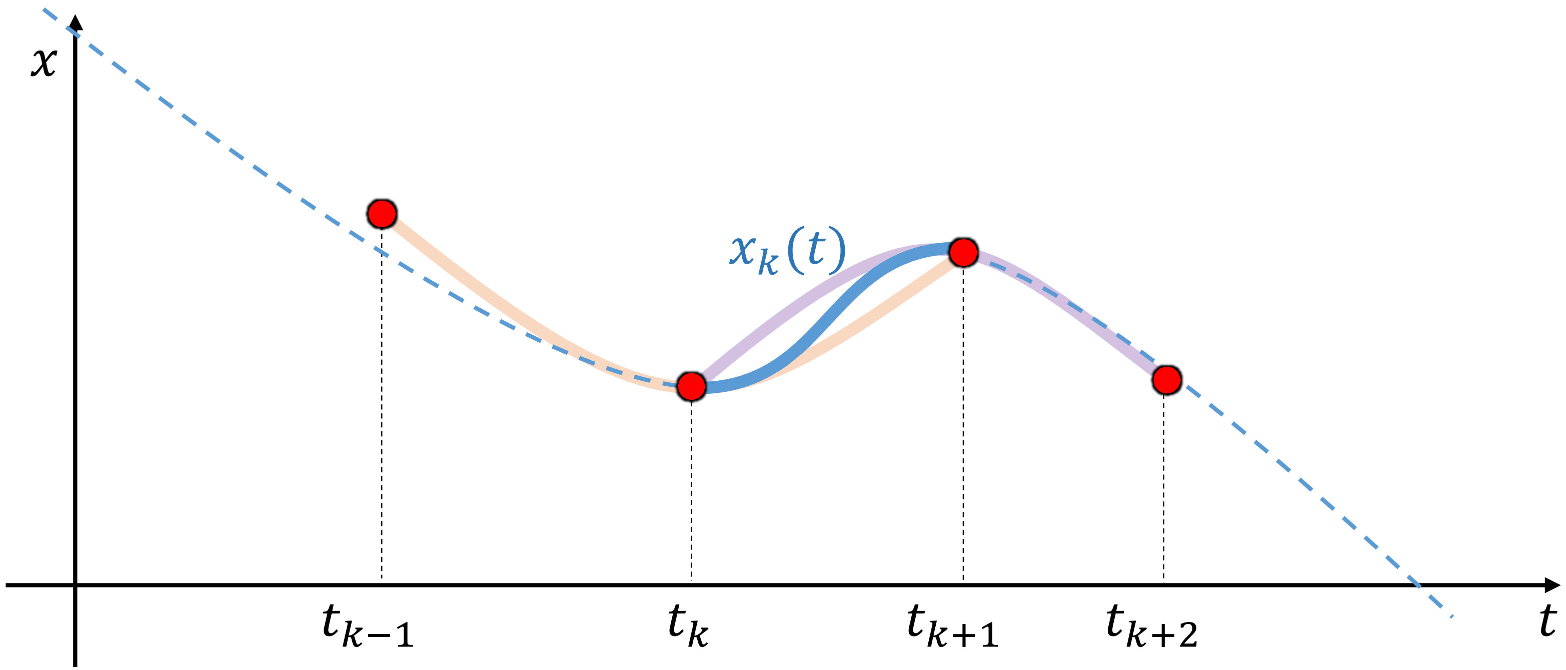
  - 3点 $(t_{k-1}, x_{k-1}), (t_k, x_k), (t_{k+1}, x_{k+1})$ を通る



3次Catmull-Romスプライン：ステップ3

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

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



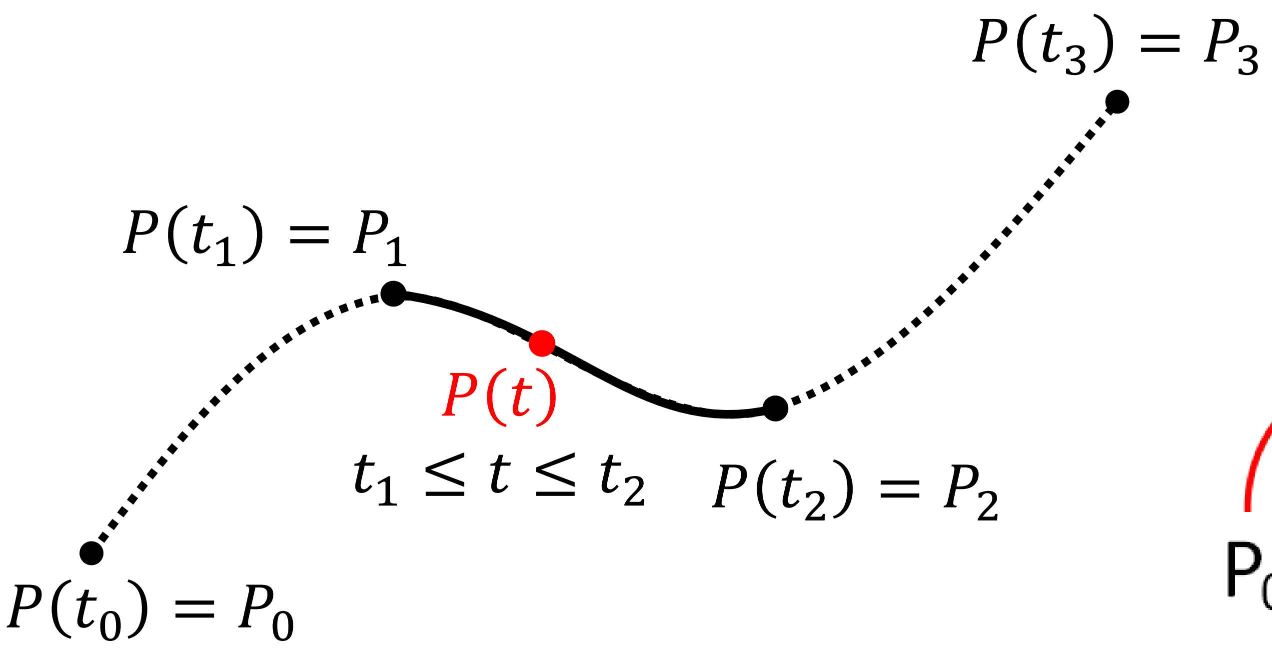
3次Catmull-RomスプラインのC¹連続性

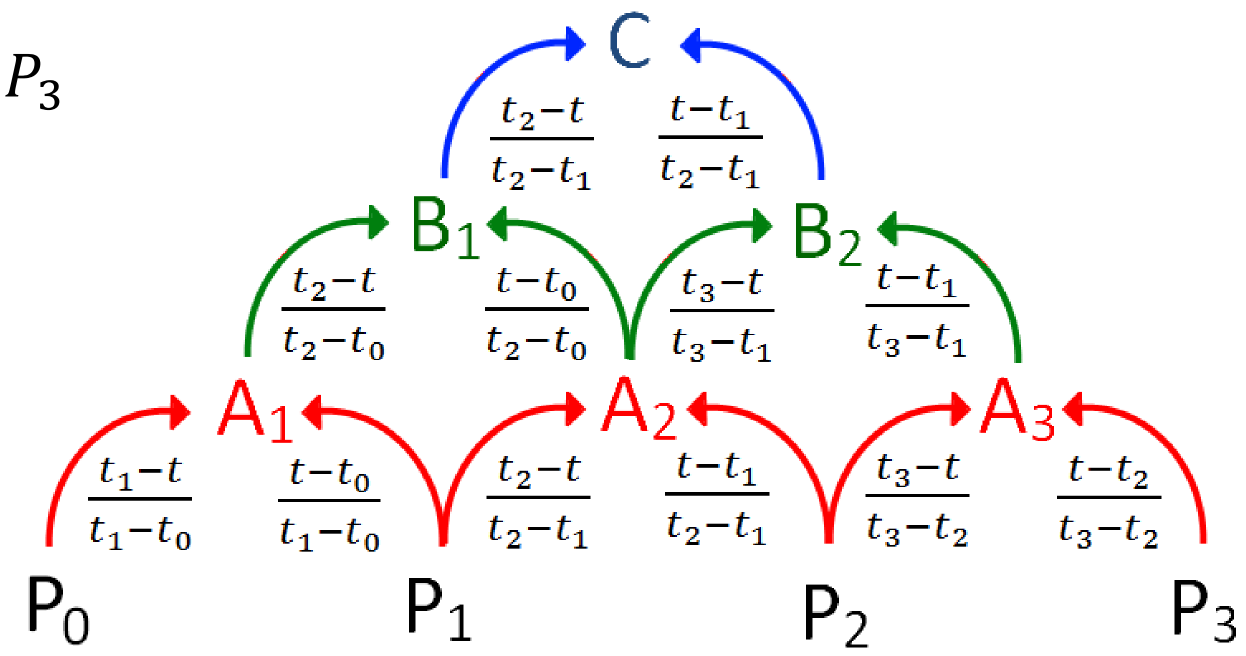
- $x_k'(t_k) = q_k'(t_k)$ となることを確認：
  - $x_k(t) = \left(1 - \frac{t-t_k}{t_{k+1}-t_k}\right) q_k(t) + \frac{t-t_k}{t_{k+1}-t_k} q_{k+1}(t)$
  - $x_k'(t) = \frac{1}{t_{k+1}-t_k}\left(-q_k(t) + q_{k+1}(t)\right) + \left(1 - \frac{t-t_k}{t_{k+1}-t_k}\right) q_k'(t) + \frac{t-t_k}{t_{k+1}-t_k} q_{k+1}'(t)$

  $t = t_k$を代入：
  - $x_k'(t_k) = \frac{1}{t_{k+1}-t_k}(-\underbrace{q_k(t_k)}_{x_k} + \underbrace{q_{k+1}(t_k)}_{x_k}) + \left(1 - \underbrace{\frac{t_k-t_k}{t_{k+1}-t_k}}_{0}\right) q_k'(t_k) + \underbrace{\frac{t_k-t_k}{t_{k+1}-t_k}}_{0} q_{k+1}'(t_k)$
- $x_k'(t_{k+1}) = q_{k+1}'(t_{k+1})$ も同様に成立

➔ $x_k(t)$ と $x_{k+1}(t)$ は値と微分が滑らかに繋がる！

3次Catmull-Romスプラインの計算方法

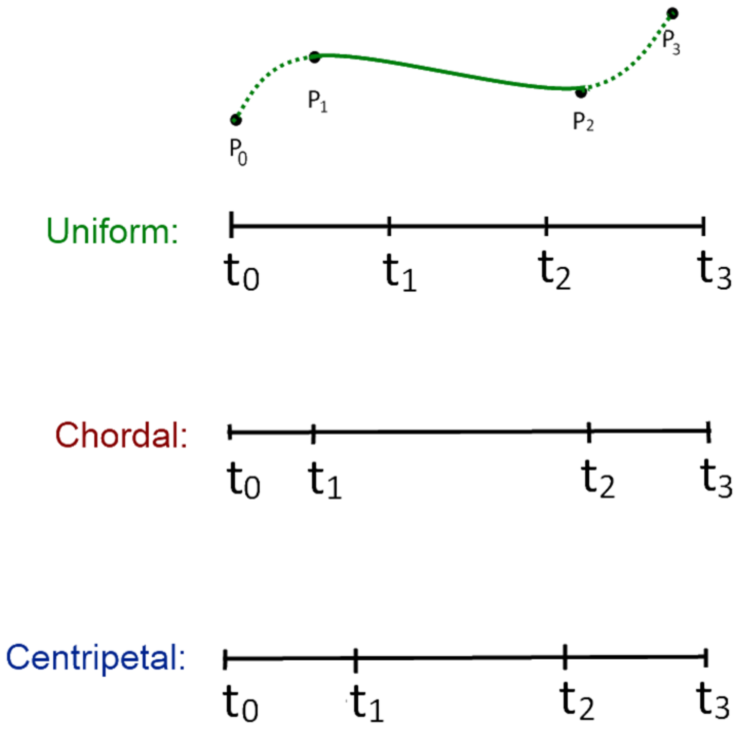


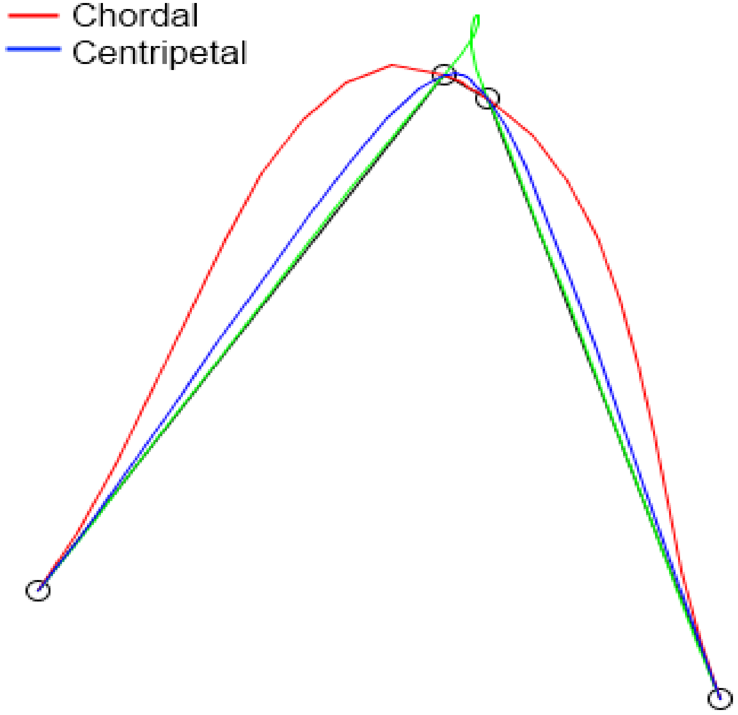


A recursive evaluation algorithm for a class of Catmull-Rom splines [Barry,Colgman,SIGGRAPH88]

パラメタ区分 t_k (ノット列) の決め方

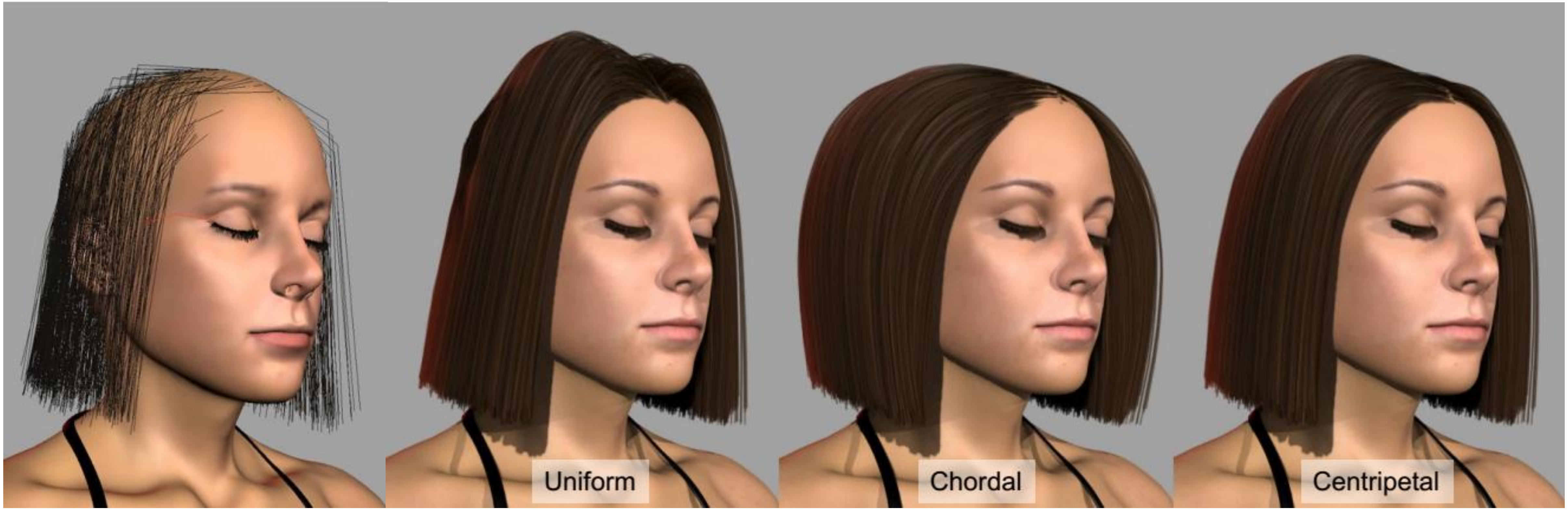
- $t_0 = 0$
- Uniform (一様)
  $$t_k = t_{k-1} + 1$$
- Chordal (弧長に基づく？)
  $$t_k = t_{k-1} + |P_{k-1} - P_k|$$
- Centripetal (求心性？)
  $$t_k = t_{k-1} + \sqrt{|P_{k-1} - P_k|}$$





Parameterization of Catmull-Rom Curves [Yuksel,Schaefer,Keyser,CAD11]

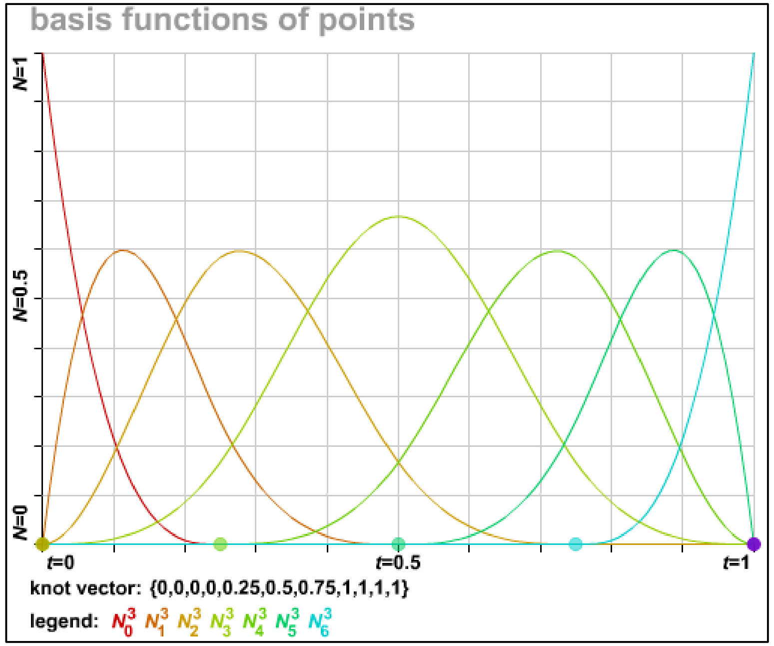
3次Catmull-Romスプラインの応用

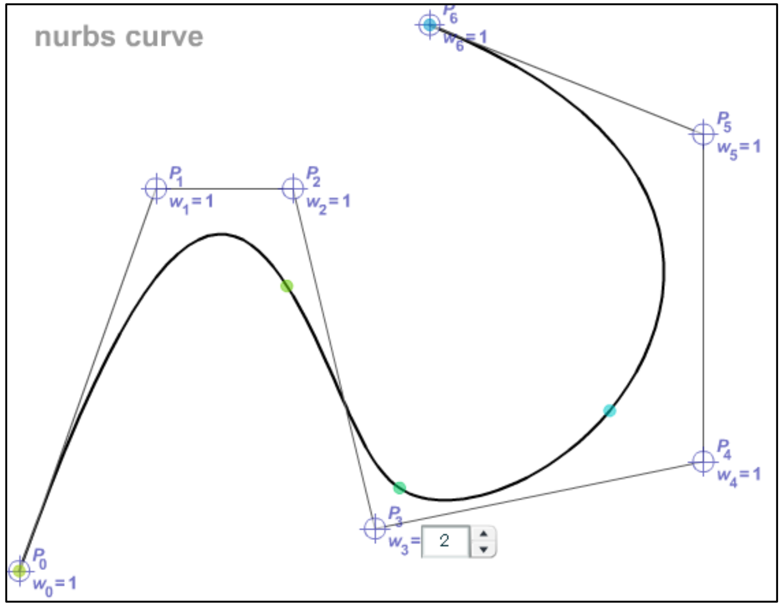


Parameterization of Catmull-Rom Curves [Yuksel,Schaefer,Keyser,CAD11]

BスプラインとNURBS

- B (**B**asis) スプライン：
  - 多項式スプラインを表すもう一つの方法
  - 曲線を基底関数の重ね合わせで表す
  - 3次の基底関数が実用上一般的
- **N**on-**U**niform **R**ational **B**-**S**pline
  - Non-Uniform：ノット列の間隔が一様でない
  - Rational：制御点が重みを持つ (有理式曲線)
- 説明はややこしいので略
  - 秀逸なFlashデモ：
    http://geometrie.foretnik.net/files/NURBS-en.swf

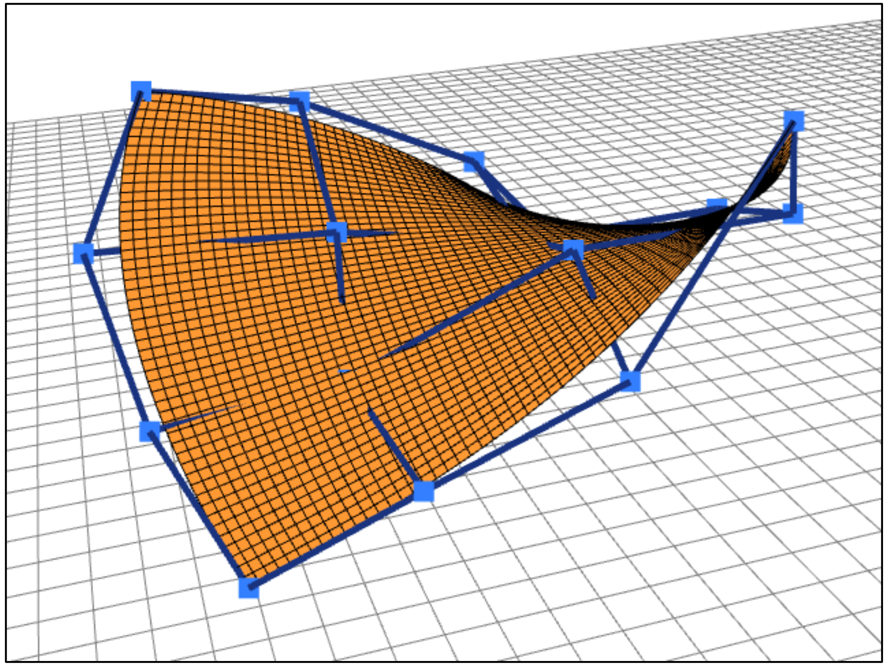




パラメトリック曲面

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

- 3次ベジェ曲面：
  - 入力：4×4=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

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

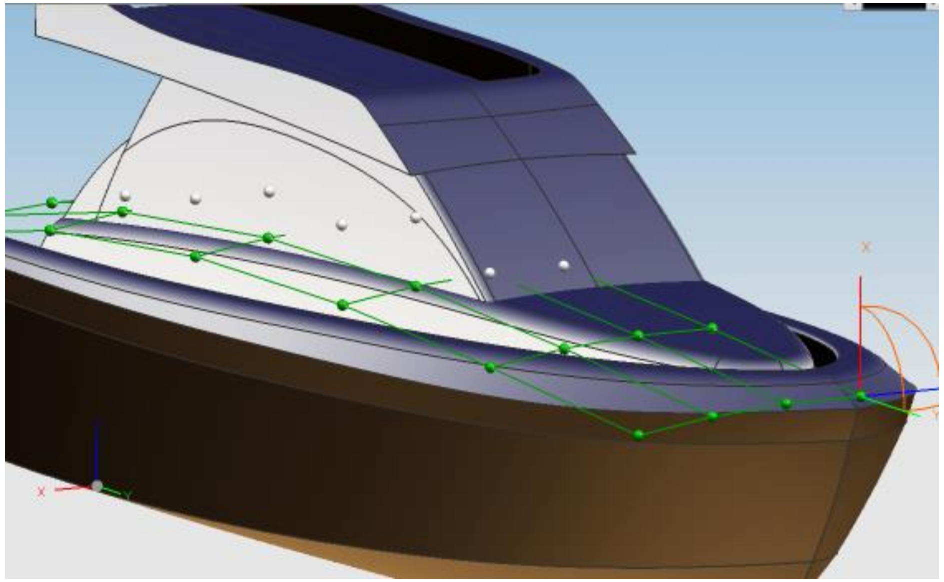
$b_1^3(t) = 3t(1-t)^2$

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

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

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