

Křivky a plochy I

Petr Felkel

Katedra počítačové grafiky a interakce, ČVUT FEL
místnost KN:E-413 na Karlově náměstí
E-mail: felkel@fel.cvut.cz

S použitím materiálů Bohuslava Hudce, Jaroslava Sloupa

Modelování křivek a ploch



Osnova

- Použití křivek a ploch v grafice a cíle při jejich návrhu
- Způsoby zápisu křivky / plochy
- Interpolační metody modelování křivek / ploch
 - Polynomiální křivky
 - Interpolace kubickými oblouky, spline
 - Fergusonovy (Hermitovské) křivky
- Aproximační metody modelování křivek
 - Bézierovy křivky / plochy
 - Coonsovy křivky / plochy
 - NURBS křivky / plochy

Křivky a plochy v počítačové grafice



Použití křivek a ploch v grafice

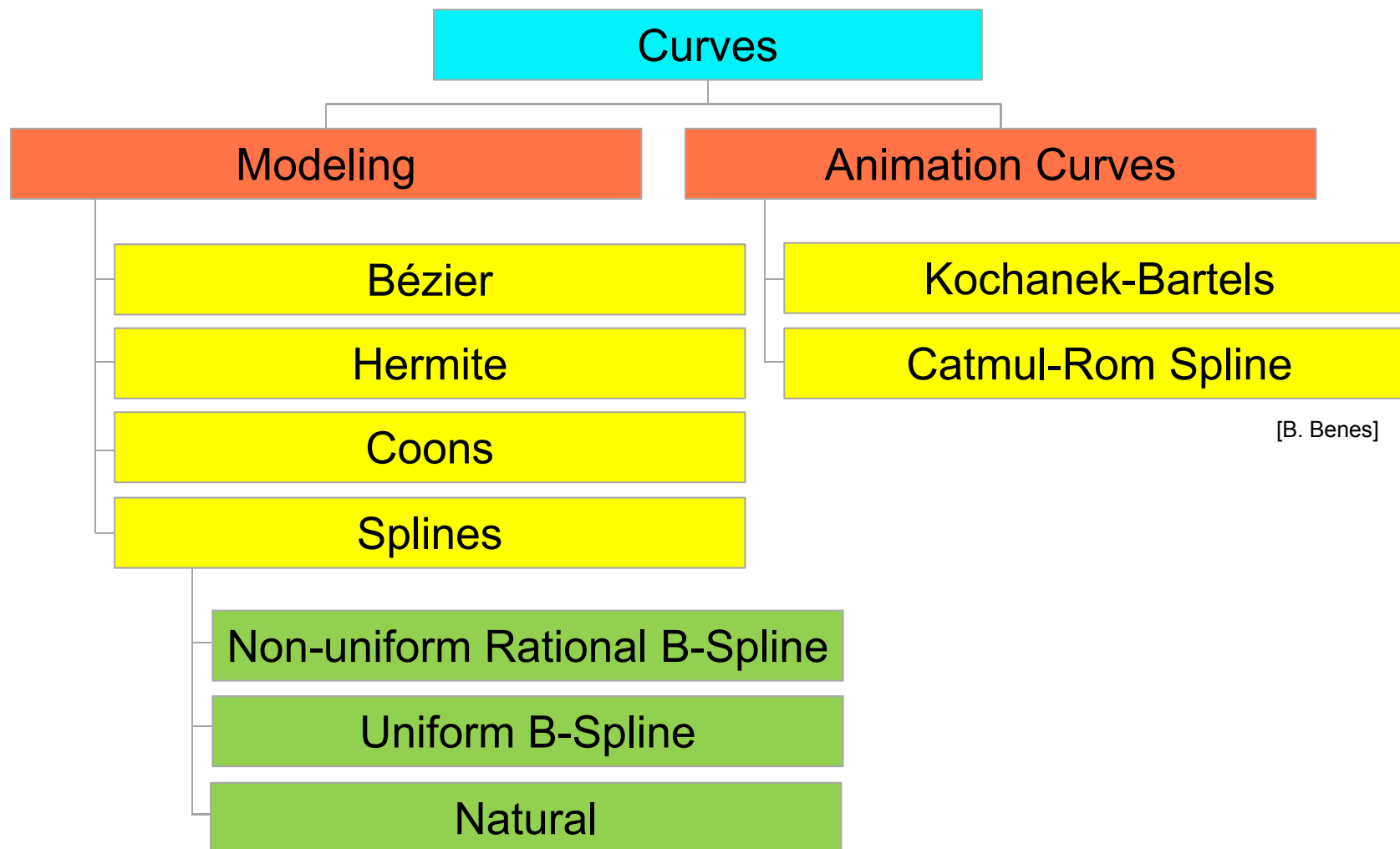
- modelování objektů (auta, průmyslový design,...) (NURBS,...)
 - definice fontů (Bézier)
 - dráha kamery a objektů při animaci (Kochanek-Bartels, Catmull-Rom)
- Co aplikace, to jiné požadavky
 - pravidelnost kroku po křivce
 - spojitost s okolními segmenty křivky
 - Intuitivní deformovatelnost
 - škála tvarů,...
 - Rozvoj v 60 letech 20 století v automobilovém průmyslu
 - de Casteljau – Citroen
 - Béziere – systém UNISURF, Renault

Hlavní cíle při návrhu křivek a ploch

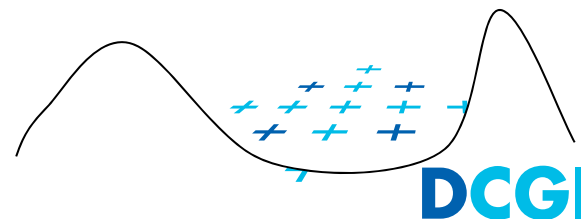


- snadná editace
 - lokalita editace tvaru
 - předvídatelné chování
 - stabilita (malé změny na vstupu způsobí malé změny na výstupu)
 - omezení konvexní obálkou řídicích bodů
 - hladkost a spojitost
 - snadný výpočet derivací
 - jednoduché vykreslování
 - invariance k afinním transformacím
- „odstínění uživatele od matematiky“
- schopnost vyjádřit jedním popisem co největší škálu tvarů (NURBS – jádro modelářů)

Taxonomie křivek v počítačové grafice



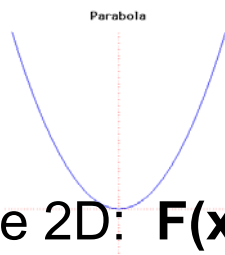
Způsoby zápisu křivky ve 2D



Explicitní zápis křivky ve 2D: $y = f(x)$, $x \in \mathbb{R}$ x nezávislá proměnná
 y závislá proměnná,
 \Rightarrow křivka je funkcí x (předpis výpočtu y pro dané x)

Př. Parabola

$$y = ax^2 + bx + c$$



Explicitní rovnice je vhodná pro prosté funkce (jedna hodnota y pro zadané x) bez svislých tečen

Implicitní zápis křivky ve 2D: $F(x,y) = 0$...definuje testovací funkci

$F(x, y) = 0 \Rightarrow$ bod $[x,y]$ leží na křivce

\Rightarrow nevhodný pro vykreslování

\Rightarrow nelze ve 3D (průsečík 2 rovnic ploch $F_1(x, y, z) \cap F_2(x, y, z)$)

Navíc se často definuje pojem uvnitř a vně křivky (nalevo, napravo)

$F(x, y) < 0 \Rightarrow$ uvnitř

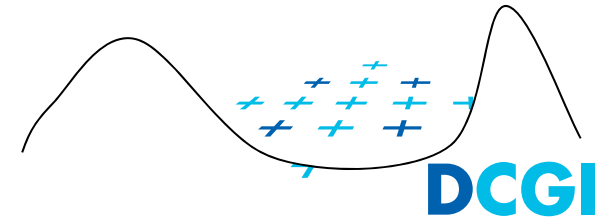
$F(x, y) > 0 \Rightarrow$ vně

Př. Kružnice se středem v bodě $[3,4]$ a poloměru 5

$$(x-3)^2 + (y-4)^2 - 5^2 = 0$$

Implicitní zápis je vhodný pro testování, zda bod leží na křivce

Zápis křivky ve 3D



Parametrický zápis křivky: (stejný pro všechny dimenze!)

- 2D křivka: $\mathbf{x} = \mathbf{x}(t)$, $\mathbf{y} = \mathbf{y}(t)$,
- 3D křivka: $\mathbf{x} = \mathbf{x}(t)$, $\mathbf{y} = \mathbf{y}(t)$, $\mathbf{z} = \mathbf{z}(t)$, parametr $t \in [\alpha, \beta]$ čas

Parametrický zápis křivky lze chápat jako pohyb bodu v závislosti na čase t .

Parametrický zápis se používá pro vykreslování, protože snadno vypočteme libovolný bod na křivce (bod je jednoznačně určen hodnotou parametru t).

Užitečné vlastnosti:

- Tečna vznikne derivací uvedených funkcí podle t
- Vhodné pro libovolný tvar křivky

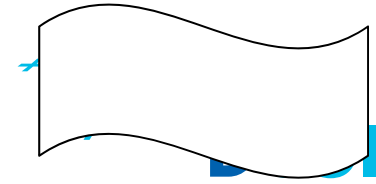
Příklad: Kružnice o poloměru r se středem v počátku:

$$x = r \cdot \cos(t), \quad y = r \cdot \sin(t), \quad 0 \leq t \leq 2\pi$$

Parametrický zápis jako sloupcová matice:

$$Q(t) = [x(t), y(t)]^T, \quad \text{resp.} \quad Q(t) = [x(t), y(t), z(t)]^T.$$

Způsoby zápisu *plochy*



Explicitní rovnice plochy: $z = f(x,y)$, $x, y \in \mathbb{R}$
 \Rightarrow plocha je funkce nezávislých proměnných x a y

POZOR
Ve 3D nejde křivka popsat implicitně ani explicitně (jen jako průnik dvou ploch)

Implicitní zápis plochy: $F(x,y,z) = 0$

Příklad: Kulová plocha o poloměru r se středem v počátku

$$x^2 + y^2 + z^2 - r^2 = 0$$

Parametrický zápis plochy:

$$\mathbf{x} = \mathbf{x}(u,v), \mathbf{y} = \mathbf{y}(u,v), \mathbf{z} = \mathbf{z}(u,v),$$

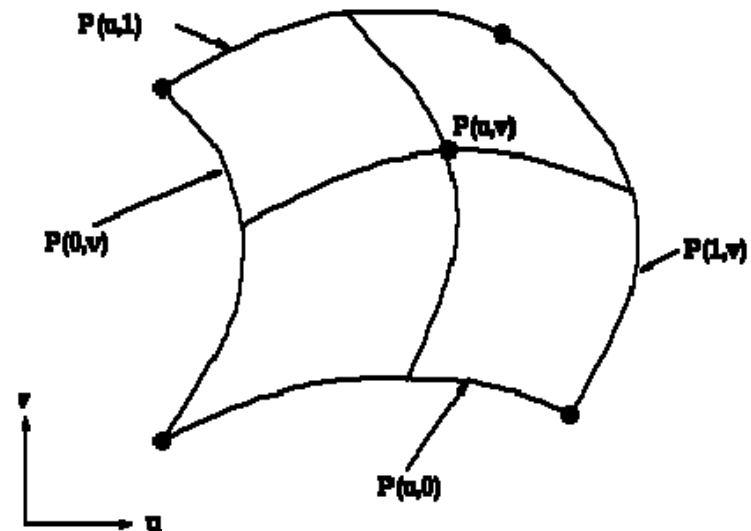
s parametry $u \in \langle \alpha, \beta \rangle, v \in \langle \gamma, \delta \rangle$

Příklad: Kulová plocha o poloměru r se středem v počátku

$$x(u,v) = r \cdot \sin(\pi u) \cdot \cos(2\pi v)$$

$$y(u,v) = r \cdot \sin(\pi u) \cdot \sin(2\pi v)$$

$$z(u,v) = r \cdot \cos(\pi u) \quad u, v \in 0..1$$



Příklad parametrické plochy
 $P(u,v), u, v \in 0..1$

Parametrické křivky



Parametrický zápis křivky

- Není jednoznačný (stejná křivka lze různě parametrizovat)

Př. pro úsečku P0P1 $L(P0,P1) = P0 + u(P1-P0),$ $u = [0...1]$
 $L(P0,P1) = (P1+P0)/2 + v(P1-P0)/2$ $v = [-1...1]$

- V grafice se používají polynomy stupně 3
- **Stupeň křivky, polynomu (degree) $n = \max$ mocnina parametru t**
- V OpenGL řád křivky (order) $k = n+1 =$ počet řídicích bodů

$$Q(t) = \sum_{i=0}^n a_i \cdot t^i, \text{ kde } a_i = \begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix}$$

- t je parametr z intervalu 0..1
- a_i je $n + 1$ vektorů koeficientů $a_i = [a_{xi}, a_{yi}, a_{zi}]^T$

Parametrický zápis polynomu stupně 4



$$Q(t) = \sum_{i=0}^n a_i \cdot t^i, \quad \text{kde} \quad a_i = \begin{bmatrix} a_{xi} \\ a_{yi} \\ a_{zi} \end{bmatrix}$$

- $Q(t) = [x, y, z]^T$, kde x, y, z jsou vzájemně nezávislé
=> 3 rovnice ve 3D
- Rozepsáno do 3 rovnic

$$Q_x(t) = a_{x0}t^0 + a_{x1}t^1 + a_{x2}t^2 + a_{x3}t^3 + a_{x4}t^4$$

$$Q_y(t) = a_{y0}t^0 + a_{y1}t^1 + a_{y2}t^2 + a_{y3}t^3 + a_{y4}t^4$$

$$Q_z(t) = a_{z0}t^0 + a_{z1}t^1 + a_{z2}t^2 + a_{z3}t^3 + a_{z4}t^4$$

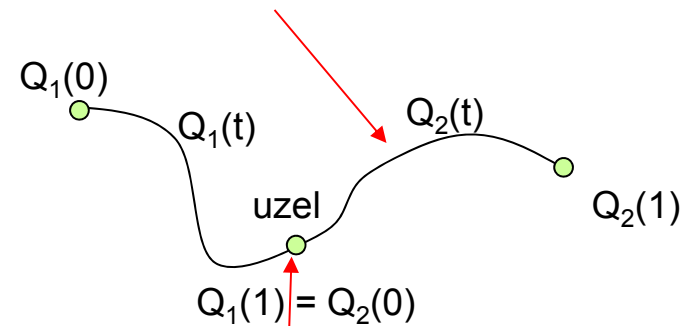
Navazování křivek



Navazování křivek

Křivka $Q(t)$ může být poskládána z částí (segmentů) $Q_1(t), Q_2(t), \dots$

Uzel (knot) = bod styku segmentů,
 $Q_1(1) = Q_2(0)$,



Navazování křivek



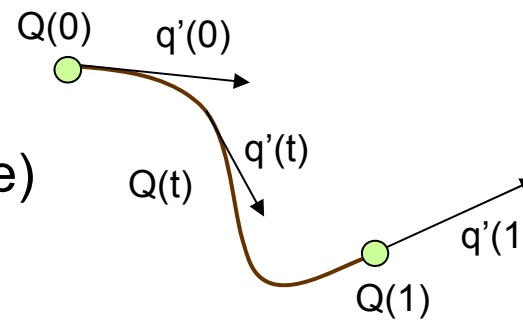
Notace

Jednosegmentová křivka

$Q(t)$ Křivka (bod křivky pro dané t)

$q'(t)$ Tečný vektor ke křivce (1. derivace)

$q''(t)$ Druhá derivace v bodě $Q(t)$

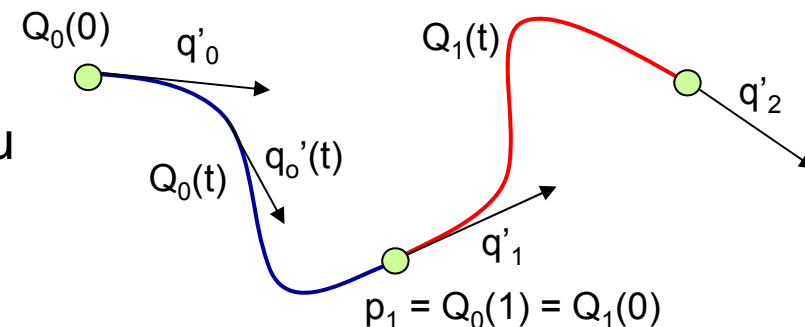


Křivka z několika segmentů

$Q_0(t)$ Nultý segment křivky,

$Q_i(t)$ i -tý segment křivky

$q_i'(t)$ Tečný vektor k i -tému segmentu



$Q_i(1) = Q_{i+1}(0)$ Uzlové body křivky

p_j Uzlové body interpolační křivky

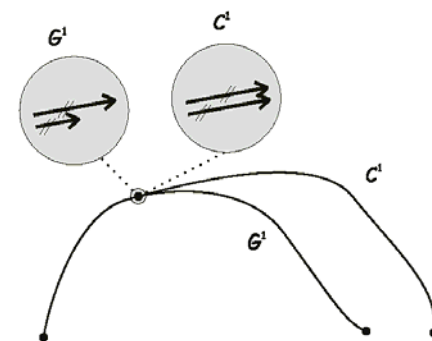
q_j' Tečný vektor (1. derivace) v uzlu j

Navazování křivek



Spojitost křivky

- **parametrická C^n**
má ve všech bodech **spojité derivace** podle parametru t řádu $0..n$ (**totožné vektory**)
Př.: C^1 má **totožné vektory tečen** příchozího a odchozího segmentu a společný koncový bod (je zároveň C^0 spojitá)
- **geometrická G^n**
má **lineárně závislé vektory n -tých derivací** příchozího a odchozího segmentu a je G^{n-1} spojitá,
tj. stejný směr a orientace, ale různá délka vektorů



Parametrická spojitost vyžaduje kromě spojitosti geometrie i spojitost zvolené parametrizace

Navazování křivek



Parametrická spojitost C^n v uzlu

C^0 ... totožný konec Q_1 a počátek Q_2

- bod se pohybuje po spojitě dráze
- směr pohybu, rychlost a zrychlení se může změnit skokem ... $f(t) = |t-0.5|$

C^1 ... totožné i vektory tečen

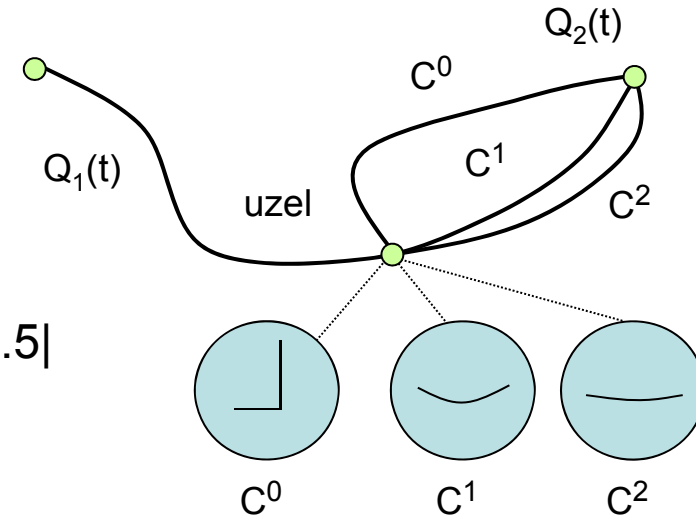
- stejný směr pohybu a rychlost
- zrychlení se může změnit skokem

C^2 ... stejné i zrychlení

Platí: $C^{n+1} \Rightarrow C^n$

... tj., spojitost C^{n+1} implikuje spojitosti $C^0, C^1, C^2, \dots, C^n$

Př.: pohyb: C^2 či C^3 , proudění kapalin: nespojitost C^3 či C^4 turbulence



[MPG] B. Benes

Navazování křivek



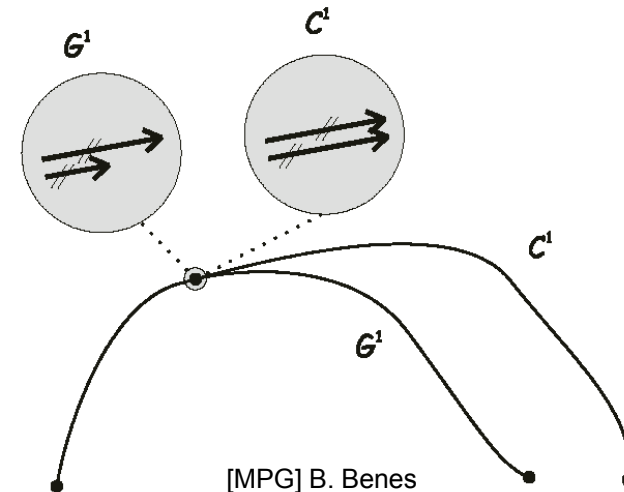
Geometrická spojitost G^n v uzlu

G^0 ... totožný koncový a počáteční bod

- jako C^0

G^1 ... souhlasně kolineární vektory tečen

- křivka vizuálně hladká
- bod nezmění směr, ale může skokem změnit rychlost a zrychlení



G^2 ... bod nezmění směr ani rychlost,
ale může skokem změnit zrychlení

Platí: $G^{n+1} \Rightarrow G^n$

... tj., spojitost G^{n+1} implikuje spojitosti $G^0, G^1, G^2, \dots, G^n$

Modelování křivek a ploch

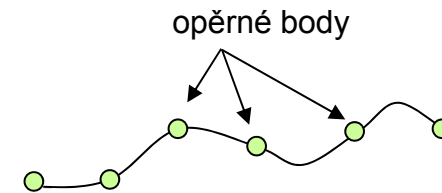


Modelování křivek

- Vstup:
 - Posloupnosti bodů (opěrné body $Q_i(\{0, 1\})$) nebo řídicí body p_i)
 - příp. derivace (tečné vektory) v bodech

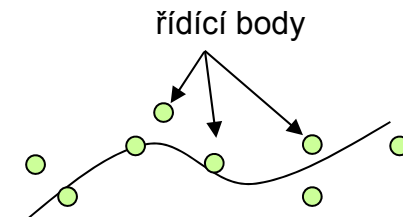
- **Interpolace**

- Křivka zadanými body musí procházet
- **Opěrné body** $Q_i(0)$ a $Q_i(1)$



- **Aproximace**

- Křivka zadanými body nemusí procházet
- **Řídicí body** p_i určují tvar křivky.



Příklad: Vozík na křivce



Funkce **alignObject()**
v asteroidech

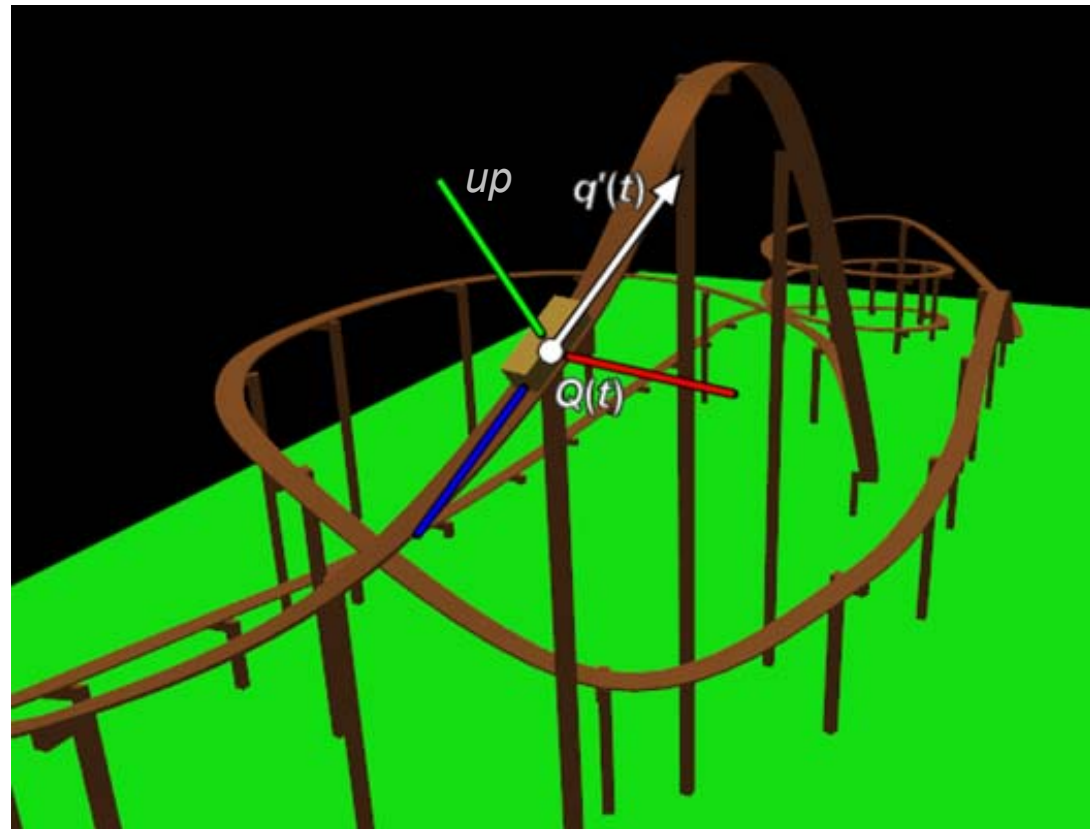
Transformuje do tečné
soustavy ke křivce

$q(t)$ = pozice vozíku

$q'(t)$ = směr jízdy vozíku

$$\vec{q}'(t) = \left[\frac{\partial x(t)}{\partial t}, \frac{\partial y(t)}{\partial t}, \frac{\partial z(t)}{\partial t} \right]$$

Stejná, jako LookAt(), ale
bez inverze matice
(je to modelová matice)



up vektor volíme dle typu aplikace

Příklad: Vozík na křivce



```
glm::mat4 alignObject(glm::vec3& position, const glm::vec3& front,
const glm::vec3& up) {

    glm::vec3 z = -glm::normalize(front);

    if (isVectorNull(z))
        z = glm::vec3(0.0, 0.0, 1.0);

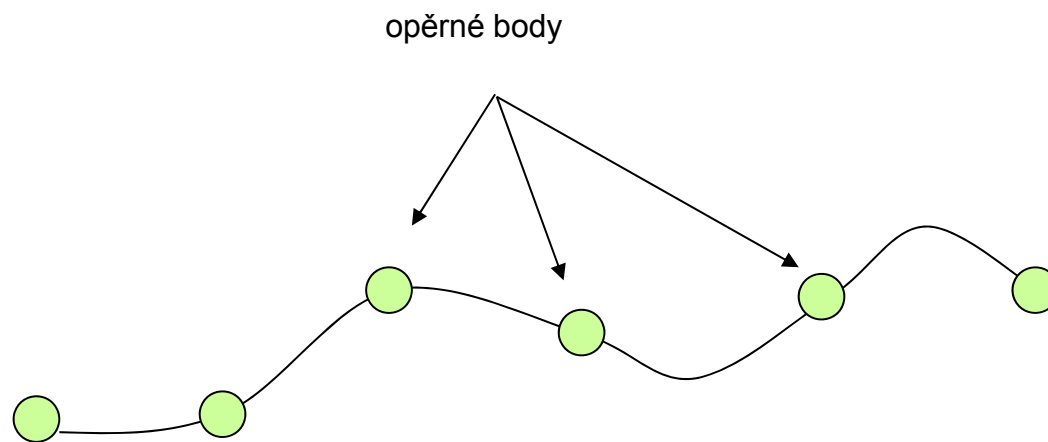
    glm::vec3 x = glm::normalize(glm::cross(up, z));

    if (isVectorNull(x))
        x = glm::vec3(1.0, 0.0, 0.0);

    glm::vec3 y = glm::cross(z, x);
    //mat4 matrix = mat4(1.0f);
    glm::mat4 matrix = glm::mat4(
        x.x,      x.y,      x.z,      0.0, // po sloupcích!!!
        y.x,      y.y,      y.z,      0.0, //
        z.x,      z.y,      z.z,      0.0,
        position.x, position.y, position.z, 1.0
    );

    return matrix; // bez inverze, tečná soustava ke křivce
}
```

Interpolační křivky

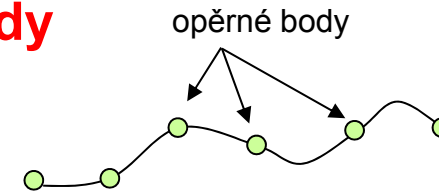


Interpolační křivky



Interpolace – křivka prochází zadanými body

Křivka zadána:



1. **Explicitně:** Souřadnice $[x, y, z]$ bodů, body se dosadí:
 - a) do **jediné rovnice polynomu** vysokého stupně
=> Vyřeší se soustava rovnic (jedna velká)
 - b) do **sady rovnic oblouků stupně 3**
=> Vyřeší se sada menších soustav rovnic
2. **Parametricky:** popořadě oblouky mezi dvojicemi bodů (Fergusson)
zadávají se dva body a dvě derivace v nich.

U rozdělení na oblouky - 1b) a 2 záleží na spojitosti napojení
snadno se dosáhne C^1 a G^1 – viz dále
zaručíme-li spojitost C^2 , vznikne *SPLINE* (st.3=> *kubický spline*)
(obecně spojitost C^n , pro křivky stupně $n+1$)

Interpolační křivky



1a) Interpolace algebraickým polynomem (explicitně - **jednou fci**)

Dáno $n + 1$ opěrných bodů $p_i = [x_i, y_i]$, $i = 0, 1, \dots, n$

- sestrojíme polynom $f^n(x)$ stupně nejvýše n
$$y_i = f^n(x_i) = a_n x_i^n + a_{n-1} x_i^{n-1} + \dots + a_1 x_i + a_0, \quad \text{kde } i = 0, 1, \dots, n$$
- souřadnice bodů dosadíme do polynomu \Rightarrow soustava $n + 1$ algebraických rovnic pro výpočet koeficientů polynomu
- vyřešením získáme koeficienty a_i

Nevýhody

- explicitní zápis křivky (jako funkce $y = f(x)$)
- \uparrow počet opěrných bodů určuje: (\uparrow = větší)
 - $\Rightarrow \uparrow$ max. **stupeň polynomu**
 - $\Rightarrow \uparrow$ operační složitost interpolace
 - $\Rightarrow \uparrow$ počet inflexních bodů – křivka je zvlněná (osciluje)
 - \Rightarrow lokální změna zasahuje celou křivku

Interpolační křivky



Příklad: Sestavte interpolační polynom procházející třemi body
 $p_0 = [0, 0]$, $p_1 = [1, 2]$, a $p_2 = [3, 3]$.

Řešení: 3 opěrné body \Rightarrow stupeň polynomu $f^n(x)$ bude $n = 2$ a proto

$$y = f^2(x) = a_2x^2 + a_1x + a_0 \quad (\text{dosadili jsme } i = 0, 1, 2)$$

Dosazením souřadnic bodů p_i do polynomu $f^2(x)$ dostaneme 3 algebraické rovnice

$$p_0: \quad a_0 = 0$$

$$p_1: \quad a_2 + a_1 + a_0 = 2$$

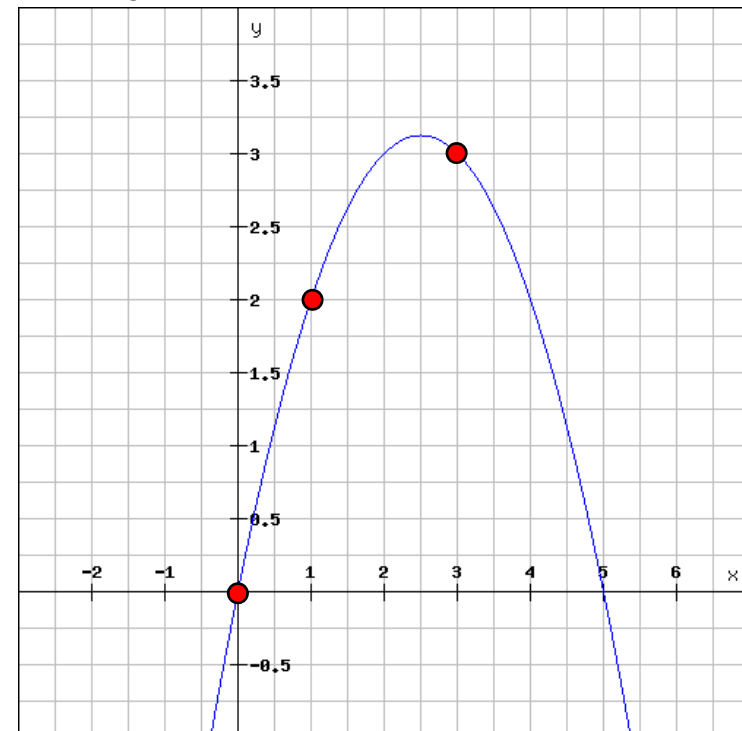
$$p_2: \quad a_2 \cdot 9 + a_1 \cdot 3 + a_0 = 3$$

Řešením těchto rovnic jsou koeficienty

$$a_2 = -0,5, \quad a_1 = 2,5 \quad \text{a} \quad a_0 = 0$$

A tedy polynom ve tvaru

$$y = -0,5x^2 + 2,5x$$

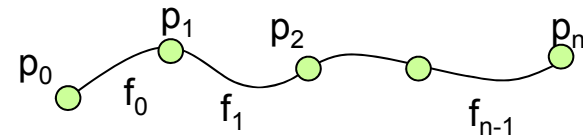


Interpolační křivky

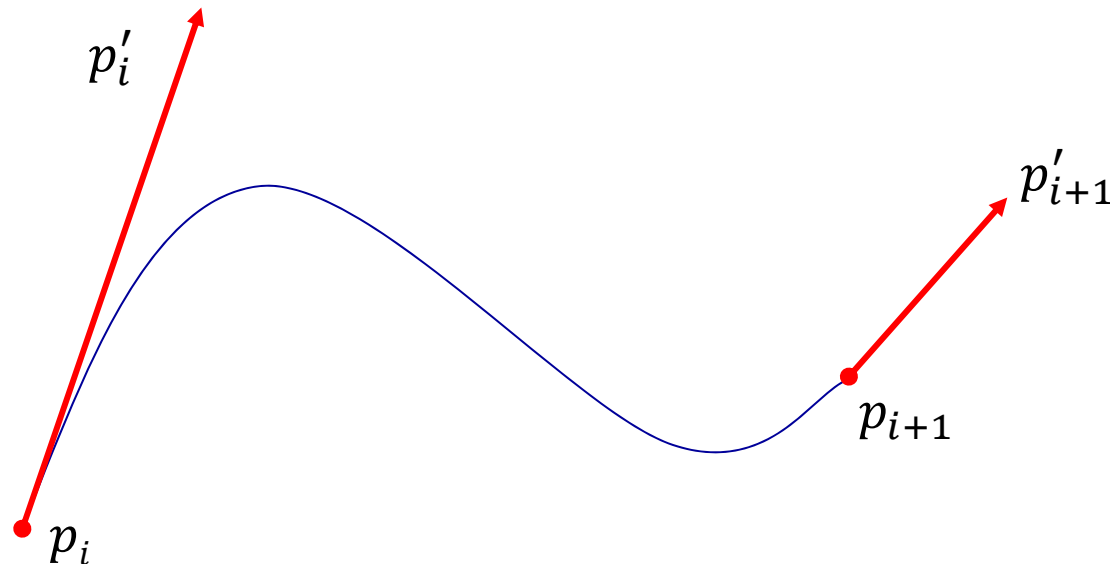


1b) Interpolace kubickými **oblouky** (explicitně - sadou funkcí st. 3)

$$f_i(x) = \sum_{j=0}^3 a_j \cdot x^j, \quad i = 0, 1, 2, \dots, 3$$



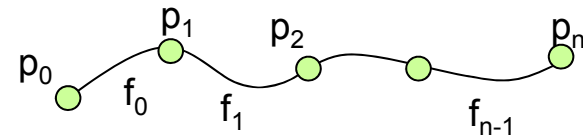
- Každý oblouk určen 2 body a 2 derivacemi
 - Počátečním a koncovým bodem (p_i a p_{i+1}) oblouku
 - Derivací v počátečním bodě $p'_i = f'_i(0)$ a koncovém bodě $p'_{i+1} = f'_i(1)$ oblouku



Interpolační křivky



... 1b) Interpolace kubickými **oblouky** (explicitně - sadou funkcí)



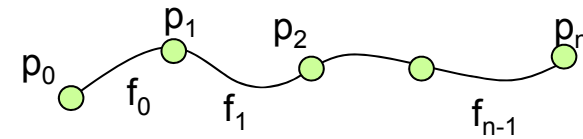
- Derivace na styku oblouků (v uzlech p_1 až p_{n-1}) lze
 - I. odhadnout středovou diferencí => spojitost C^1 (G^1)
=> Catmull-Rom spline ... viz další strana
 - II. vypočítat z derivací funkcí oblouků tak, aby byla spojitá C^2
=> přirozený spline
- Derivace na počátku p_0 a na konci p_n celé křivky se určí, obvykle $=0$
- Lokalizují se změny na sousední oblouky, ne na celou křivku

Interpolační křivky

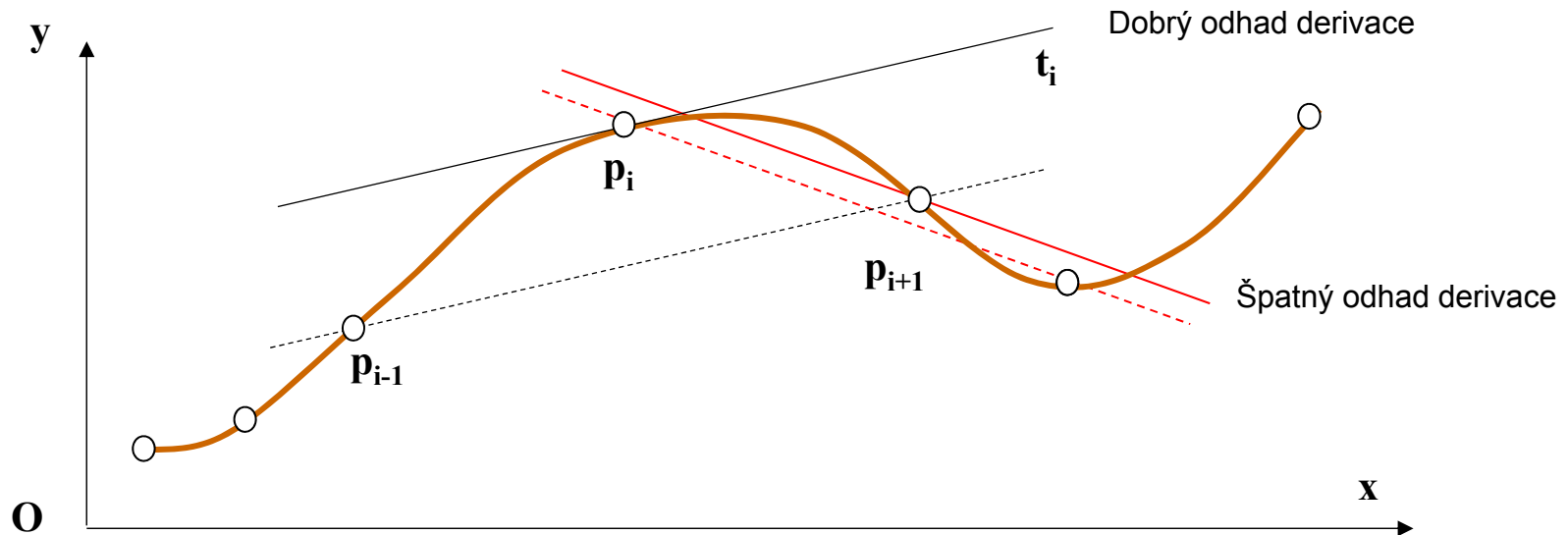


1b) I. **Derivace v uzlech p_1 až p_{n-1} se odhaduje středovou diferencí**

$$f'(x_i) = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \quad i = 1, 2, \dots, n-1$$



Metoda středových diferencí - odhad derivací v bodě P_i



$f(x)$ a $f'(x)$ jsou spojité $\Rightarrow (C^1 \text{ spojitost})$

Interpolační křivky



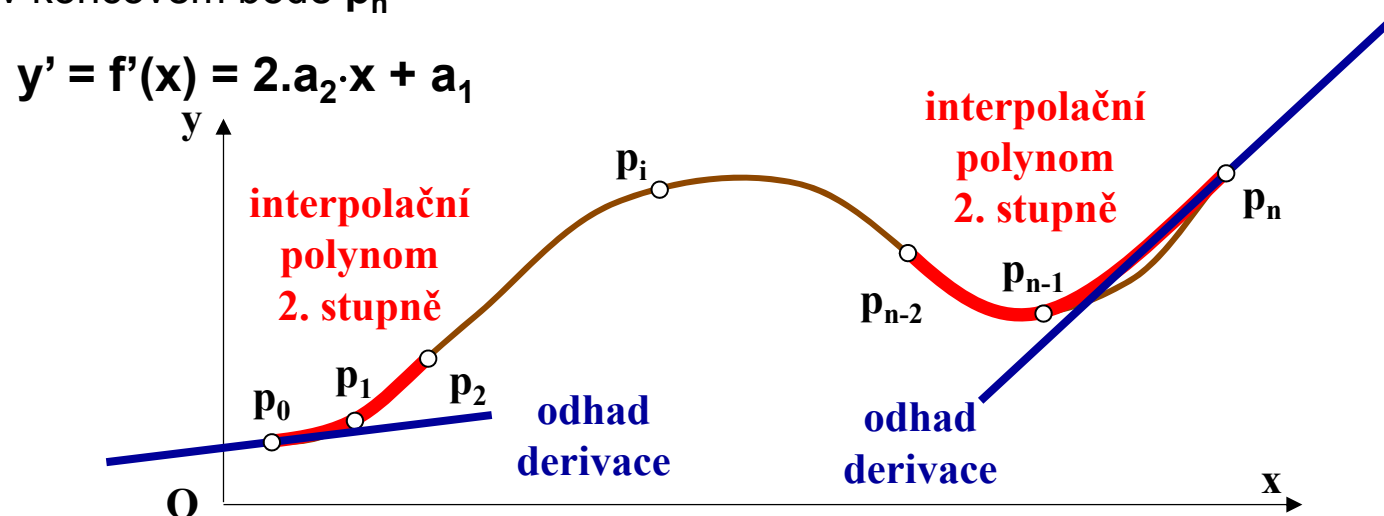
1b) I. **Derivace** v počátečním a koncovém bodě p_0 a p_n se nastaví na **0** nebo **odhadem** následujícím postupem

Odhad

- Sestrojíme dva interpolační polynomy $f(x)$ stupně 2, jeden pro 3 počáteční body (p_0, p_1, p_2) a druhý pro poslední 3 opěrné body (p_{n-2}, p_{n-1}, p_n)

$$y = f(x) = a_2 \cdot x^2 + a_1 \cdot x + a_0$$

- Derivace těchto polynomů pak použijeme jako hledané derivace v počátečním bodě p_0 a v koncovém bodě p_n

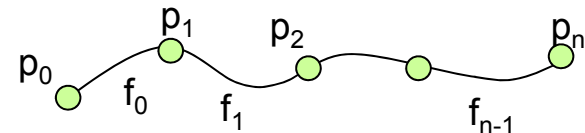


Interpolační křivky



1b) II. SPLINE – **derivace se spočítá** analyticky z oblouků $f_i(x)$

$f(x)$, $f'(x)$, $f''(x)$ jsou spojité \Rightarrow **C^2 spojitost**



Podmínky pro interpolaci spline křivky $f(x)$ **kubickými oblouky $f_i(x)$:**

- křivka prochází opěrnými body $p_i = [x_i, y_i]$, $i = 0, 1, \dots, n$ n+1 bodů

$y_i = f_i(x_i) = a_3 x_i^3 + a_2 x_i^2 + a_1 x_i + a_0$, $i = 0, 1, \dots, n-1$ počátky oblouků

$y_{i+1} = f_i(x_{i+1}) = a_3 x_{i+1}^3 + a_2 x_{i+1}^2 + a_1 x_{i+1} + a_0$, $i = 0, 1, \dots, n-1$ konce oblouků

$\Rightarrow 2n$ rovnic
- spojitost **1. a 2. derivace** v opěrných bodech

$f'_i(x_{i+1}) = f'_{i+1}(x_{i+1})$, $i = 0, \dots, n-2$

$f''_i(x_{i+1}) = f''_{i+1}(x_{i+1})$, $\Rightarrow 2(n - 1)$ rovnic
- v počátku a na konci $f''_0(x_0) = 0$ (přirozený spline)

$f''_{n-1}(x_n) = 0$ $\Rightarrow 2$ rovnice

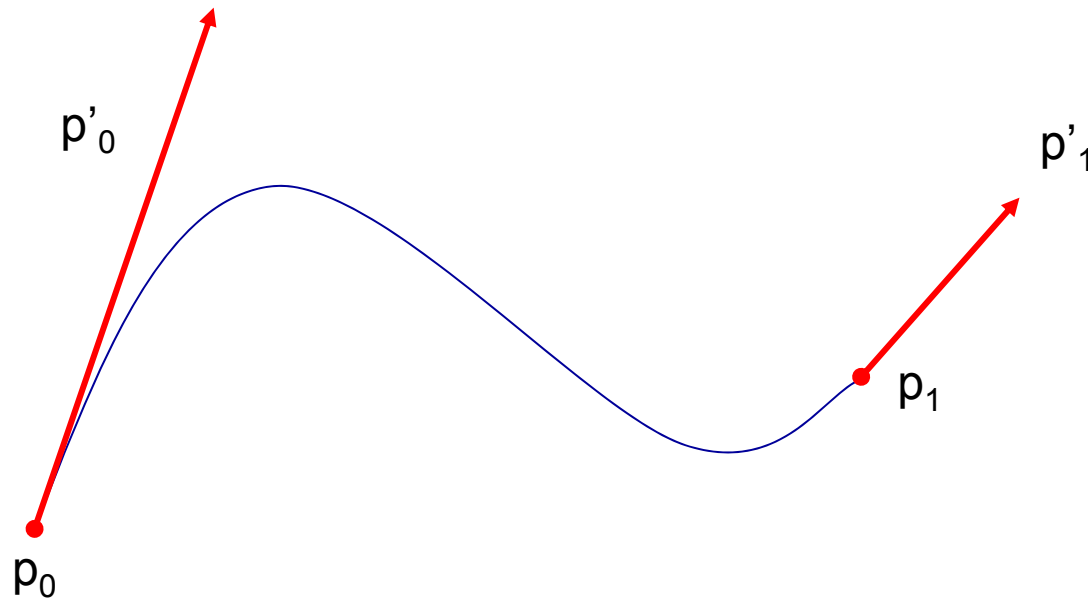
celkem: $2n + 2(n - 1) + 2 = 4n$ rovnic (pro n+1 bodů ve 2D)

Interpolační křivky



2. Parametricky - Fergusonova (Hermitovská) kubika

- zadána dvěma koncovými **body** a **derivacemi** (tangenty) v nich



p_0 počáteční bod, $t=0$

p_1 koncový bod, $t=1$

p'_0 tečna v počátečním bodě, $t = 0$

p'_1 tečna v koncovém bodě, $t = 1$

Interpolační křivky



2. Fergusonova (Hermitovská) kubika (parametricky)

$$Q(t) = p_0 \cdot F_0(t) + p_1 \cdot F_1(t) + p'_0 \cdot F_2(t) + p'_1 \cdot F_3(t), \quad t \in [0, 1]$$

$$F_0(t) = 2t^3 - 3t^2 + 1$$

$$F_1(t) = -2t^3 + 3t^2$$

$$F_2(t) = t^3 - 2t^2 + t$$

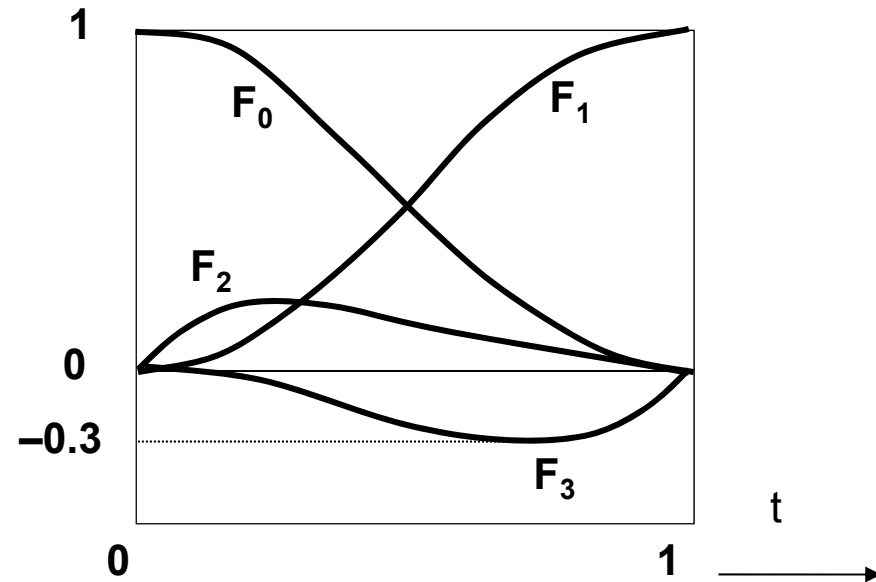
$$F_3(t) = t^3 - t^2$$

Kontrola: Dosadíme-li:

$$Q(0) = p_0$$

$$Q(1) = p_1$$

Prochází body p_0 a p_1



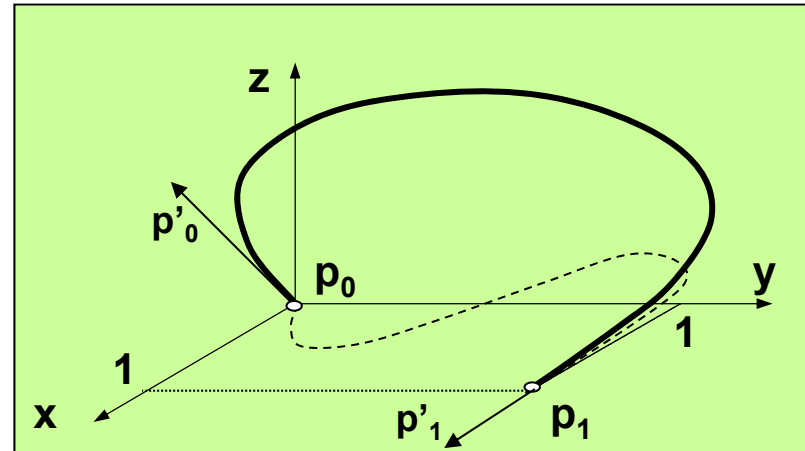
Kubické Hermitovské polynomy
blending functions

(nabývají i záporných hodnot)

Interpolační křivky

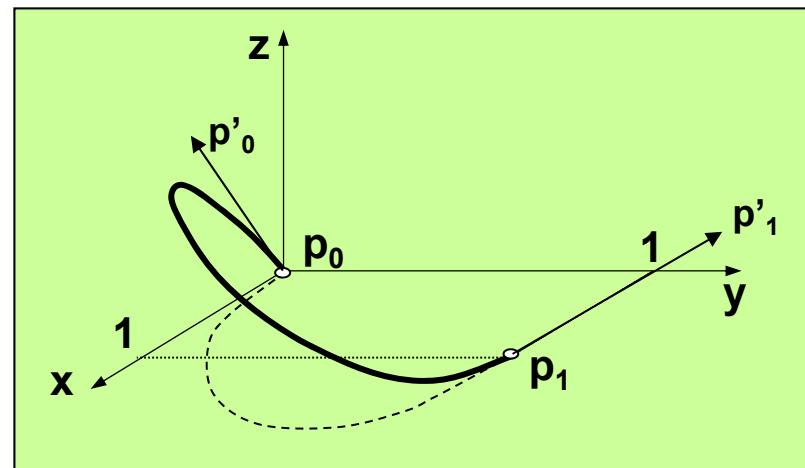
- Fergusonova kubika

$$p'_0 = [5, 0, 5]$$
$$p'_1 = [10, 0, 0]$$



- Fergusonova kubika

$$p'_0 = [3, 0, 5]$$
$$p'_1 = [-5, 0, 0]$$



POZOR! Vektory derivací jsou značně zkráceny!!!

2. Speciální případy Hermitovské (Fergusonovy) kubiky

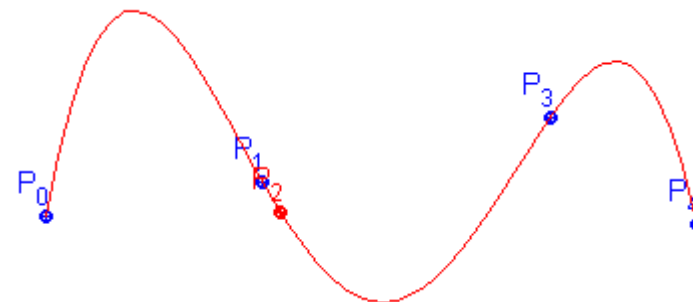
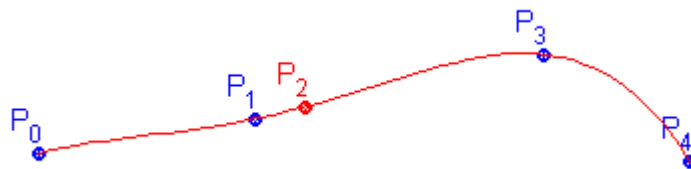
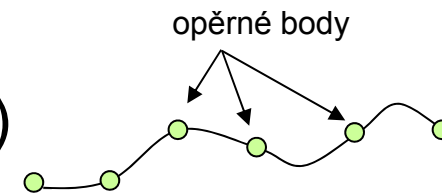
- používají se v animacích
 - jsou C^1 spojité
- **Catmull-Rom splines**
 - používá se často při animaci kamery
 - tangenty se nastavují středovou diferencí $(p_{k+1} - p_{k-1}) / 2$
 - **Kochanek-Bartels Splines**
 - Tangenty váženým průměrem vektorů k předchozímu a následujícímu bodu - parametry **b**ias, **t**ension a **c**ontinuity (spojitost)

Interpolační křivky

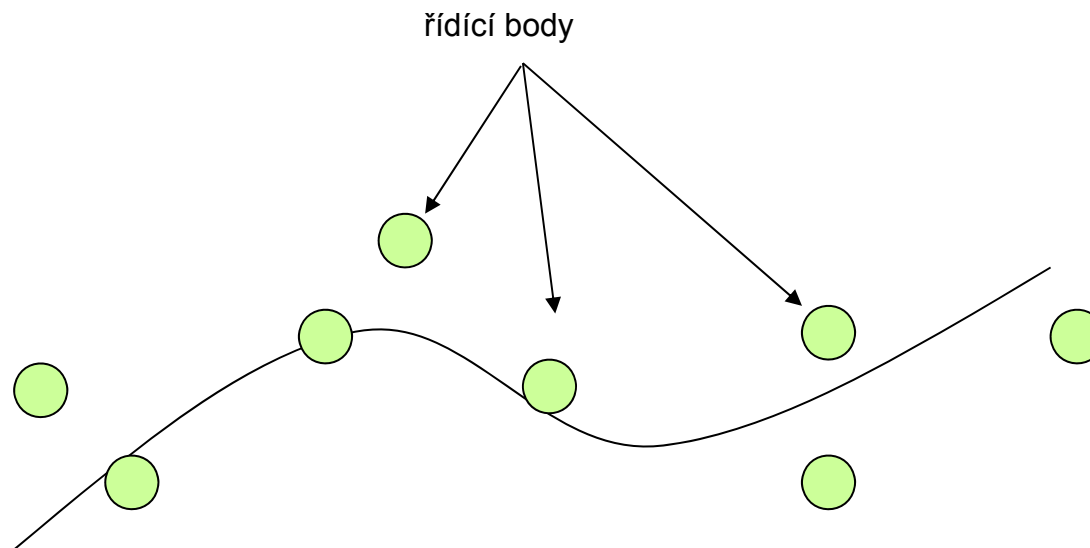


Shrnutí interpolačních metod

- Křivka musí procházet zadanými **opěrnými body**
- Obtížné splnit
- Náchylná k oscilacím
(Drobný posun bodu => obrovské změny)



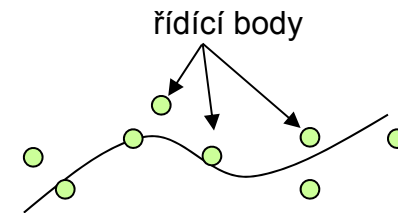
Aproximační křivky



Aproximační křivky



- Neprocházejí všemi řídicími body, některými však procházet mohou (Beziérový křivky)
- Nemusí dokonce procházet žádnými řídicími body (Coonsova křivka)



Tvar křivky určují **řídicí body** (control points) p_i

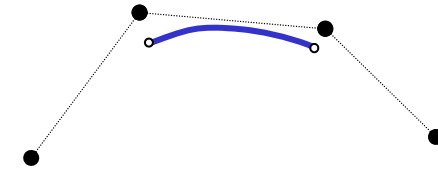
Aproximační křivky



Aproximační křivka

konvexní (barycentrická) **kombinace řídicích bodů** p_i

$$Q(t) = \sum_{i=0}^n p_i \cdot F_i(t), \quad t \in \alpha.. \beta$$



$F_i(t)$ - soustava bázových funkcí (mísících funkcí - *blending functions*)
 p_i - řídicí body

Obecné vlastnosti aproximační křivky

1. **Bázové funkce** $F_i(t)$ jsou všechny **kladné**.
 \Rightarrow modelovaná křivka se vychyluje **ve směru pohybu řídicího bodu**.
2. $\sum_{i=0}^n F_i(t) = 1$ pro všechna $t \in \alpha.. \beta$
 \Rightarrow Modelovaná křivka $Q(t)$ **leží uvnitř konvexní obálky** řídicích bodů p_i .
3. Bod p_i ovlivňuje tvar modelované křivky v celém intervalu $\alpha.. \beta$.
4. Křivka $Q(t)$ má v bodě t spojitou k -tou derivaci,
jestliže mají v tomto bodě spojitě k -té derivace bázové funkce $F_i(t)$

Aproximační metody modelování křivek



Probereme tyto aproximační křivky

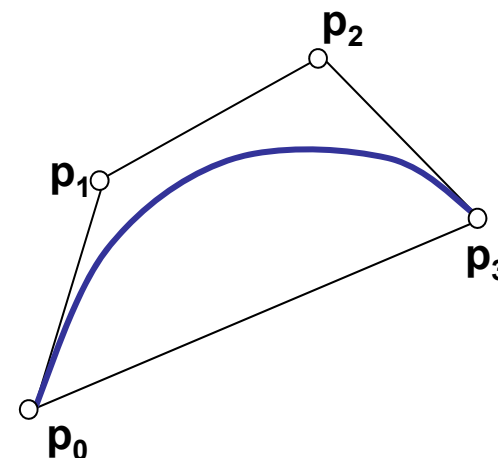
- Bézierovu
- Coonsovou
- NURBS - příště

Bézierovy křivky **n-tého** stupně



Určena $n+1$ řídicími body a vztahem

$$Q(t) = \sum_{i=0}^n p_i \cdot B_{i,n}(t), \quad t \in 0..1$$



kde

$p_i, \quad i = 0,1,2,\dots,n$ řídicí body

$B_{i,n}(t) = c_{i,n} \cdot t^i \cdot (1-t)^{n-i}$ Bernsteinovy polynomy n - tého stupně

$$c_{i,n} = \binom{n}{i} = \frac{n!}{(n-i)! i!} \quad \text{binomické koeficienty}$$

Bézierovy křivky n-tého stupně



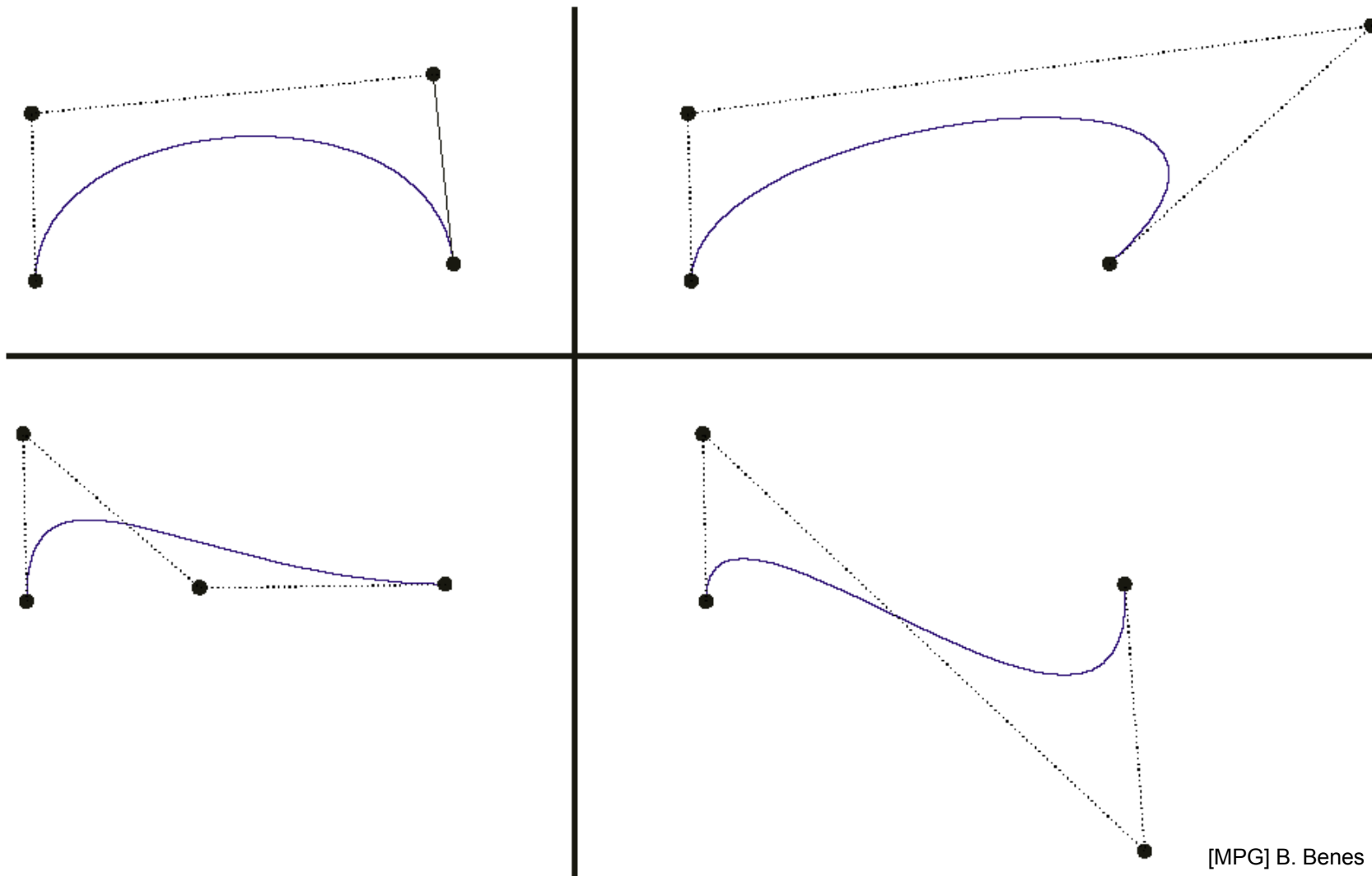
n .. stupeň

	$n = 0$	$n = 1$	$n = 2$	$n = 3$
$i = 0$	$B_{0,0}(t) = 1$	$B_{0,1}(t) = 1 - t$	$B_{0,2}(t) = (1 - t)^2$	$B_{0,3}(t) = (1 - t)^3$
$i = 1$		$B_{1,1}(t) = t$	$B_{1,2}(t) = 2t \cdot (1 - t)$	$B_{1,3}(t) = 3t \cdot (1 - t)^2$
$i = 2$			$B_{2,2}(t) = t^2$	$B_{2,3}(t) = 3t^2 \cdot (1 - t)$
$i = 3$				$B_{3,3}(t) = t^3$

↑↑
stupeň

↑
i ... index polynomu
(řídícího bodu)

Aproximační metody modelování křivek



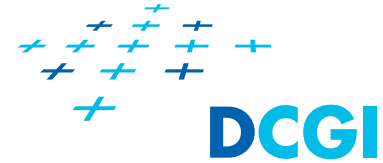
[MPG] B. Benes

Vlastnosti Bézierových křivek (řádu n)



1. **Počátečním bodem** křivky je bod p_0 , **koncovým bodem** je bod p_n .
2. Křivka se v počátečním bodě **dotýká první hrany řídicího polygonu**, v koncovém bodě se **dotýká poslední hrany řídicího polygonu**.
tečný vektor v počátečním bodě: $\mathbf{p}'_0 = n (p_1 - p_0)$
tečný vektor v koncovém bodě: $\mathbf{p}'_n = n (p_n - p_{n-1})$.
3. Bézierova křivka **leží v konvexní obálce** množiny bodů p_i .
4. **Derivace** Bézierovy křivky **je spojitá Bézierova křivka**.
5. Bézierova křivka je symetrická vůči posloupnosti řídicích bodů.
6. Řídící bod p_i má **pseudolokální vliv** na tvar Bézierovy křivky.
7. Bézierovy křivky jsou **invariantní vůči afinní transformaci** ($x \rightarrow Mx+v$).
(posunutí, otáčení, změna měřítka, zkosení, zrcadlení, paralelní projekce)
8. Jsou-li řídicí body p_0, p_1, \dots, p_n rovnoměrně rozmístěny na úsečce (p_0, p_n) , pak Bézierovou křivkou je úsečka (p_0, p_n) .
(lineární přesnost).

Vztah mezi Beziérovými a Fergusonovými křivkami



- Pro stupeň 3 je Beziérová křivka kubikou parametru t
- Fergusonova křivka je také kubikou parametru t
- Křivka prochází krajními body jak pro Beziérovu křivku, tak i Fergusonovu kubiku
- Je tedy společný vzorec polynomu třetího stupně s různými koeficienty.
- Rozdíl je v definici:
 - U Fergusonovy křivky máme zadaný tečný vektor explicitně
 - U Beziérovy kubiky je tečný vektor zadán obálkou

Bézierovy křivky n-tého stupně



$$Q(t) = \sum_{i=0}^n p_i \cdot B_{i,n}(t), \quad t \in 0..1$$

Poznámky:

1. Bernsteinovy polynomy můžeme též zapsat ve tvaru

$$B_{i,n}(t) = \sum_{j=i}^n (-1)^{j-i} \binom{n}{j} \binom{j}{i} t^j$$

2. nebo rekurentním vztahem (lineární kombinací B nižších stupňů)

$$B_{i,n}(t) = (1-t) B_{i,n-1}(t) + t B_{i-1,n-1}(t)$$

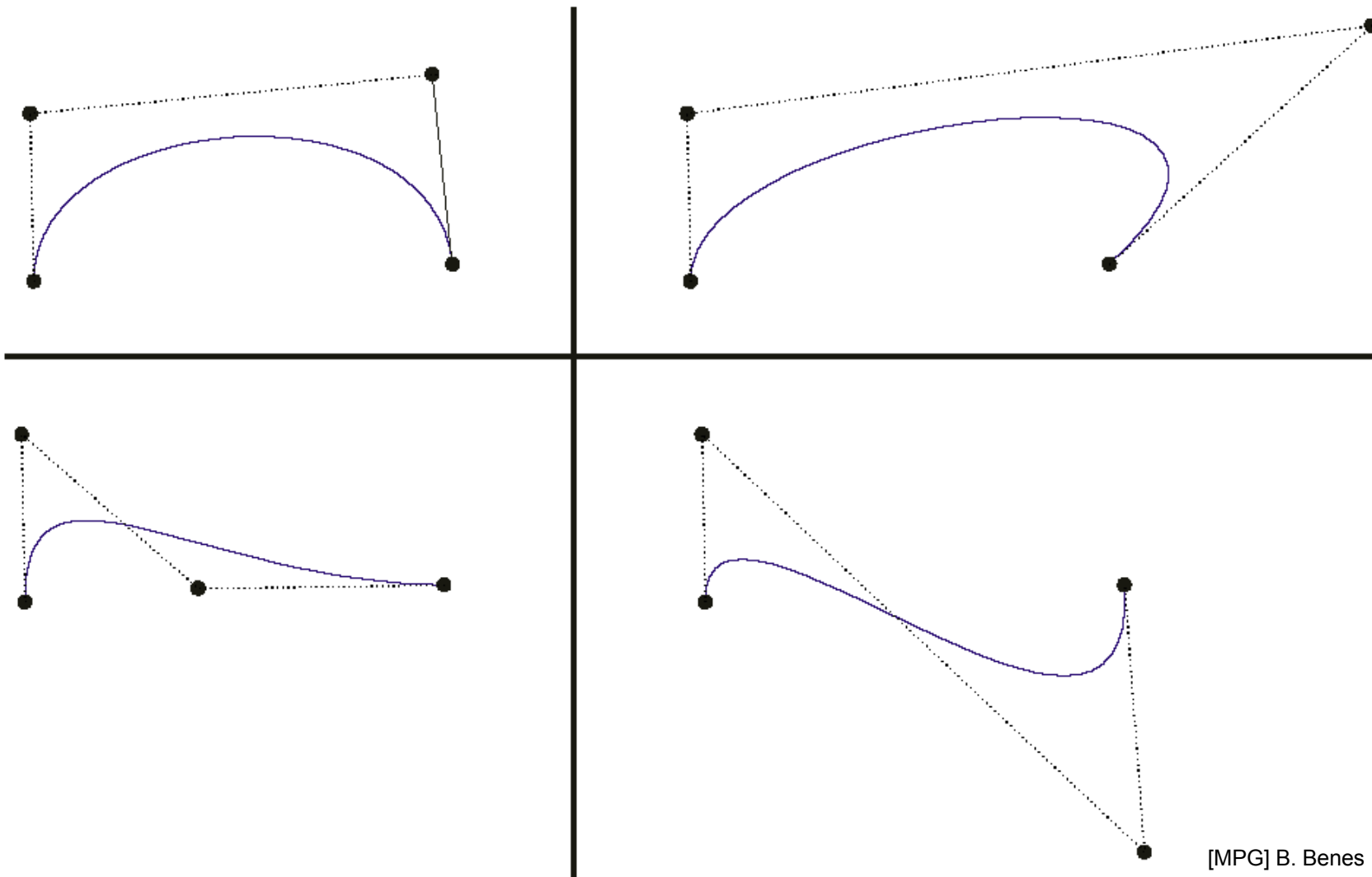
3. nebo pomocí symetrie

$$B_{i,n}(t) = (1-t) B_{i,n-i}(t)$$

4. Pro derivaci platí

$$B'_{i,n}(t) = n \cdot [B_{i-1,n-1}(t) - B_{i,n-1}(t)]$$

Bézierovy křivky n-tého stupně



[MPG] B. Benes

Bézierovy křivky n-tého stupně



Vykreslování Bézierových křivek

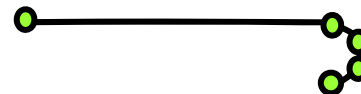
Naivní algoritmus vykreslování není efektivní

- produkuje stejně dlouhé úsečky bez ohledu na zakřivení segmentu



Algoritmus de Casteljau

- Adaptační dle zakřivení => Algoritmus de Casteljau



Vykreslení Bézierových oblouků – naivní algoritmus [Beneš]

Dán parametr t a čtveřice řídicích bodů P_0, \dots, P_3

```
PointT Q(double t,  
          PointT P0, PointT P1, PointT P2, PointT P3) {  
  
    double B0, B1, B2, B3;  
    static PointT hlp;  
  
    B0 = (1-t) * (1-t) * (1-t);  
    B1 = 3*t * (1-t) * (1-t);  
    B2 = 3*t*t * (1-t);  
    B3 = t*t*t;  
  
    hlp.x = P0.x*B0 + P1.x*B1 + P2.x*B2 + P3.x*B3;  
    hlp.y = P0.y*B0 + P1.y*B1 + P2.y*B2 + P3.y*B3;  
    return hlp;  
}
```

Vykreslení Bézierových oblouků – naivní algoritmus [Beneš]

```
Bezier(PointT P0,PointT P1, PointT P2, PointT P3, int n){
//P0,...,P3 are the control points
//n is the # of lines used for approximation
double t, deltat;
PointT a, b;
int i;

deltat = 1.0/n;    //curve will be approximated by n
lines
a = Q(0, P0, P1, P2, P3); //the first point of the curve
t = 0;
for (i=0;i<n;i++){ //for all lines do
    b = Q(t+deltat, P0, P1, P2, P3); //endpoint of line
    Line(a,b);           //render line
    a=b;                 //end becomes the start now
    t += deltat;        //increase the t variable
}
}
```

Algoritmus de Casteljau



- Založen na rekurzivní definici Bernsteinových polynomů
- Vrací jeden bod na křivce:

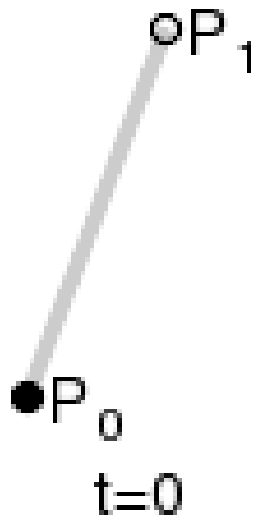
$$p_{i,j}(t) = (1-t)p_{i-1,j-1}(t) + tp_{i,j-1}(t)$$

kde $j = 1, 2, \dots, n$

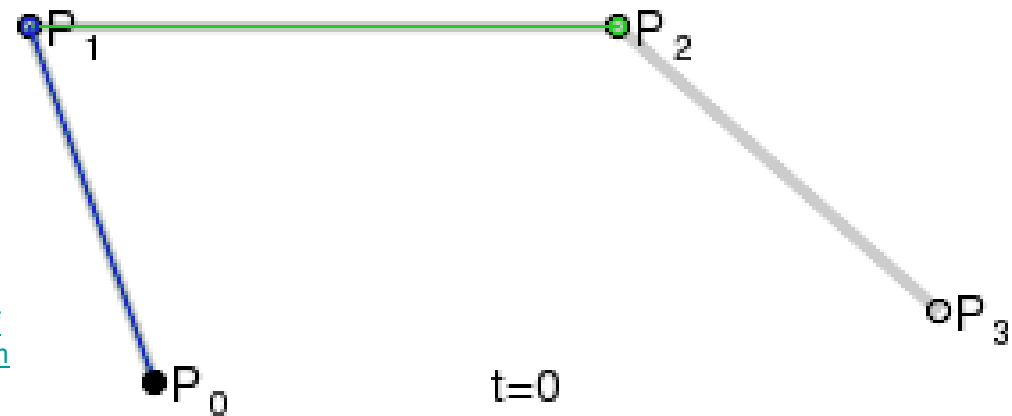
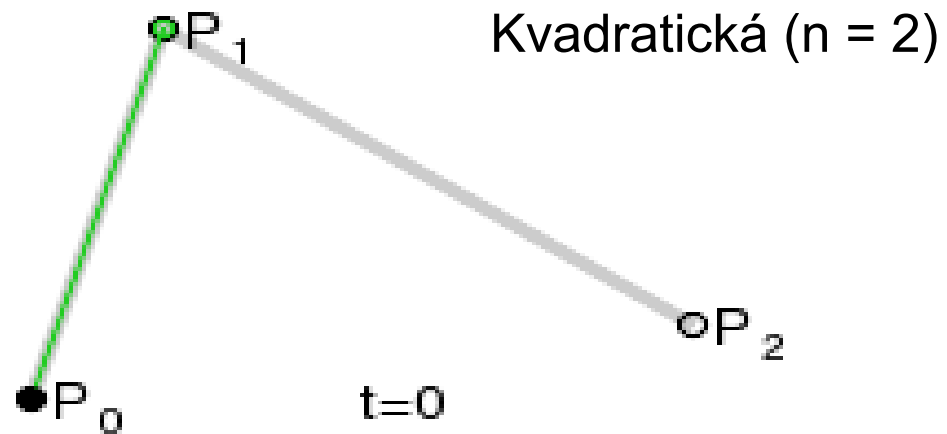
$$i = j, \dots, n$$

- Počáteční hodnoty bodů jsou $p_{i,0} = p_i$
- Koeficient j odpovídá úrovni rekurze (roste pomaleji)
- Poslední bod je $p_{n,n} = p_n$

Bézierovy křivky různých řádů



Lineární
 $n = 1$



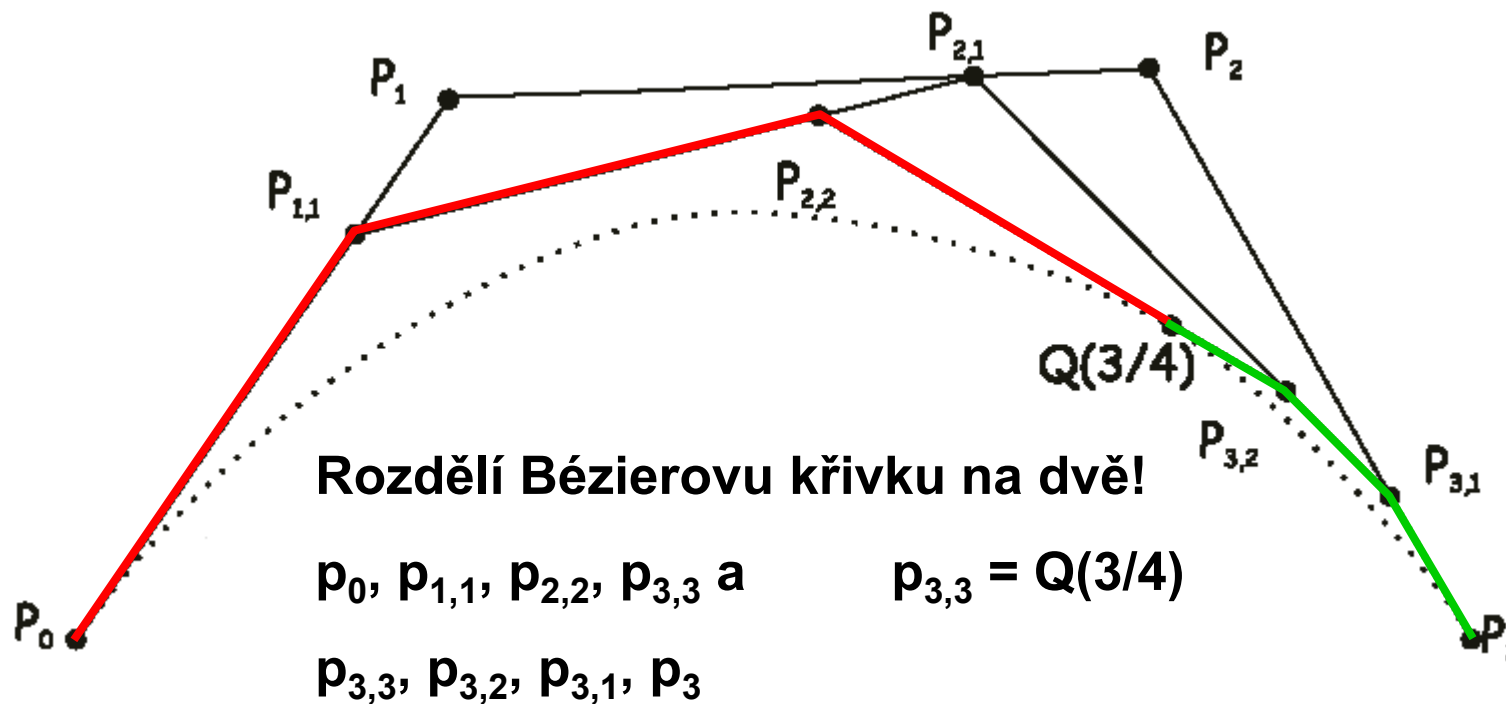
Kubická ($n = 3$)

http://commons.wikimedia.org/wiki/File:Bezier_linear_anim.gif
http://commons.wikimedia.org/wiki/File:Bezier_quadratic_anim
http://commons.wikimedia.org/wiki/File:Bezier_cubic_anim.gif

Algoritmus de Casteljau pro $t = 3/4$



Přidání řídicích bodů beze změny tvaru je možné pro libovolné t



Algoritmus de Casteljau



```
/* Adaptive rasterization of a Bézier curve */
void DeCasteljau (PointT P[])
{
    PointT left[4], right[4];
    if (FlatEnough(P))           // podmínka zastavení rekurze
    {
        Line (P[0], P[3]);
        return;
    }
    else
    {
        SplitCurve(P,left,right);
        DeCasteljau(left);
        DeCasteljau(right);
    }
}
```

[Beneš]

Pseudokód algoritmu de Casteljau pro $t = 1/2$



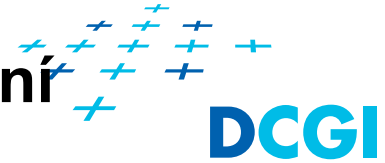
/* Input is the curve control polygon P .

Alg. **splits the curve exactly in the middle** and returns two control polygons, one for the *left* and one for the *right* segment. */

```
void SplitCurve(PointT P[], PointT left[], PointT right[]) {  
    PointT hlp;  
    left[0] = P[0];  
    left[1].x = (P[0].x+P[1].x)/2.0; hlp.x = (P[1].x+P[2].x)/2.0;  
    left[1].y = (P[0].y+P[1].y)/2.0; hlp.y = (P[1].y+P[2].y)/2.0;  
    left[2].x = (left[1].x+hlp.x)/2.0; left[2].y = (left[1].y+hlp.y)/2.0;  
    right[3]=P[3];  
    right[2].x=(P[2].x+P[3].x)/2.0; right[1].x = (right[2].x+hlp.x)/2.0;  
    right[2].y=(P[2].y+P[3].y)/2.0; right[1].y = (right[2].y+hlp.y)/2.0;  
    left[3].x= right[0].x = (left[2].x + right[1].x)/2.0;  
    left[3].y= right[0].y = (left[2].y + right[1].y)/2.0;  
}
```

[Beneš]

Kritérium pro zastavení rekurze při vykreslování



Algoritmus de Casteljau

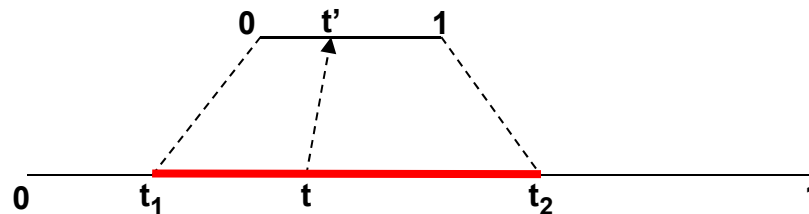
Kritérium pro zastavení rekurze při vykreslování ve FlatEnough(P)

1. Řídící body padnou do jednoho pixelu
2. Řídící body jsou menší než jeden pixel (antialiasing)
3. Řídící polygon se zúžil téměř na přímku
4. Plocha řídicího polygonu je malá
5. ...

Napojování Bézierových oblouků (segmentů) a reparametrizace

- Křivku poskládáme z Bézierových oblouků => Lokální změny při editaci
- **Celá křivka** má rozsah parametru $0 \leq t \leq 1$ (definiční obor)
- **segmenty** (oblouky) mají rozsah parametru $t_1 \leq t \leq t_2$
- nutno reparametrizovat (pro každý segment na t' z intervalu $0 \leq t \leq 1$)

$$t' = \frac{t - t_1}{t_2 - t_1}$$



- zvětší interval. $\langle t_i, t_{i+1} \rangle$ pro každý segment na $\langle 0, 1 \rangle$

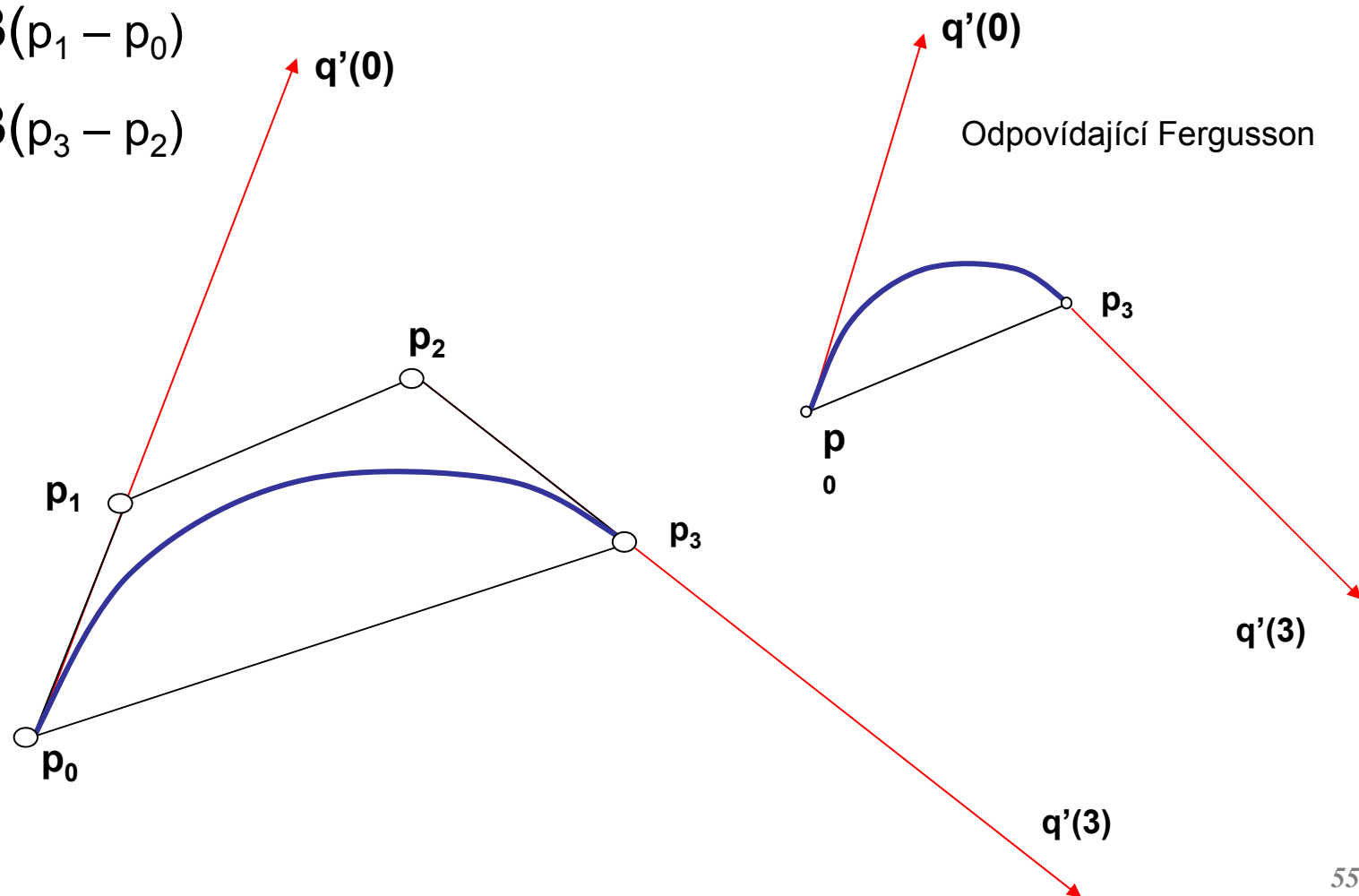
Příklad napojování Beziérových oblouků



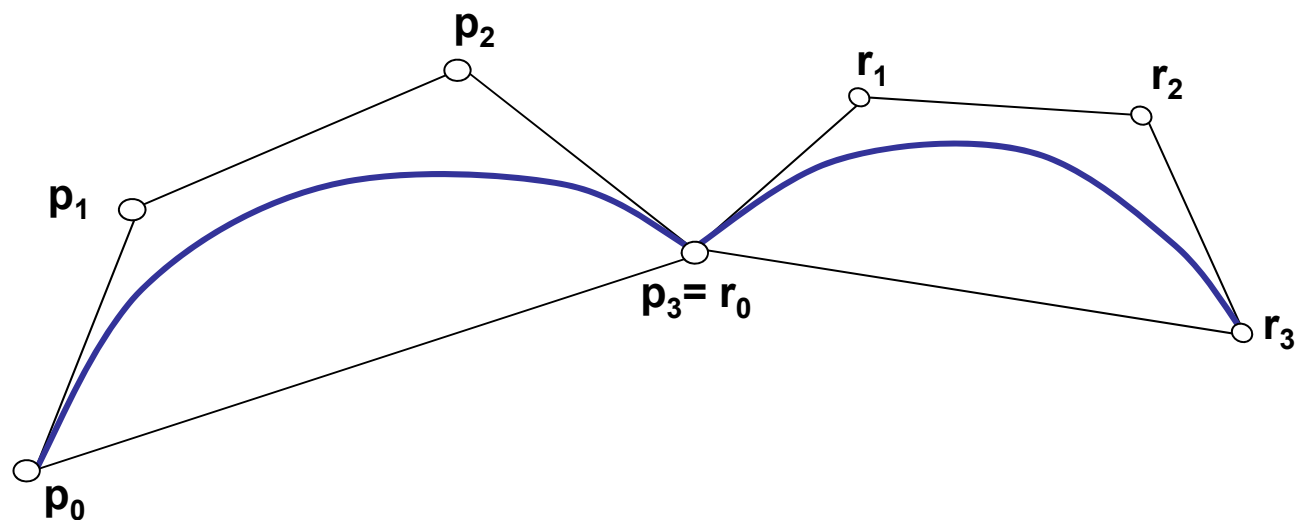
Napojování Béziových oblouků a derivace v koncových bodech

$$\mathbf{q}'(0) = 3(\mathbf{p}_1 - \mathbf{p}_0)$$

$$\mathbf{q}'(3) = 3(\mathbf{p}_3 - \mathbf{p}_2)$$



Napojování Bézierových oblouků - spojitost C^0



Spojitost C^0

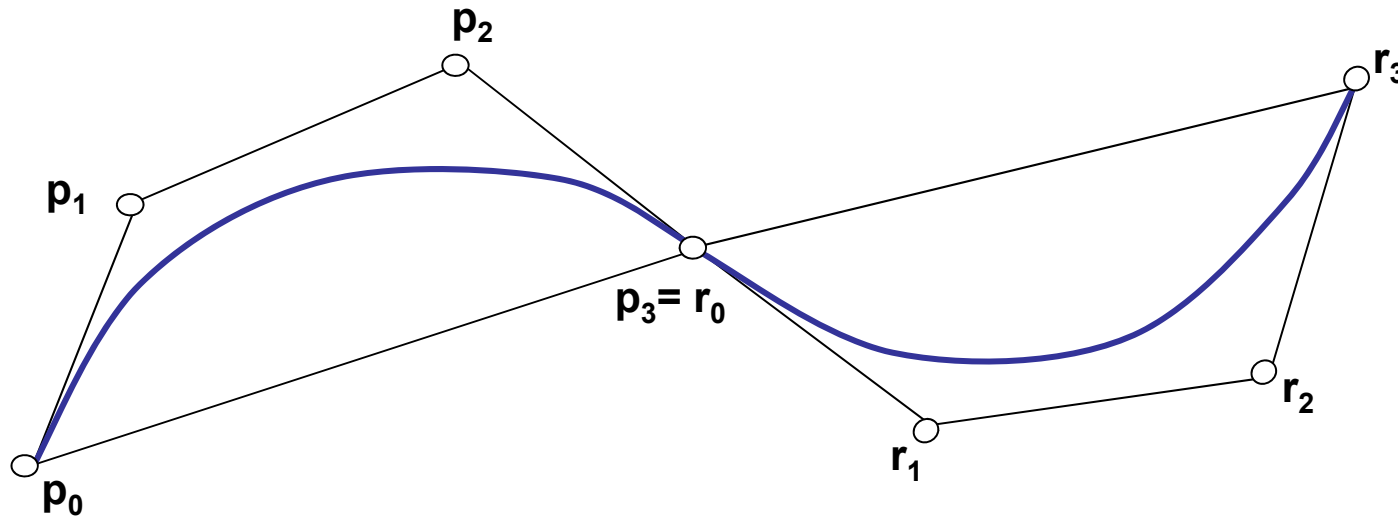
koncový bod oblouku = počáteční dalšího

$$p_3 = r_0$$

Napojování Bézierových oblouků - spojitost C^1



Spojitost C^1 (parametrická) – první derivace je shodná

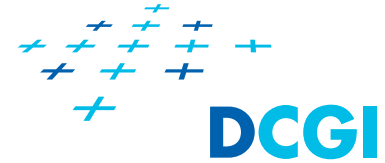


Spojitost C^1

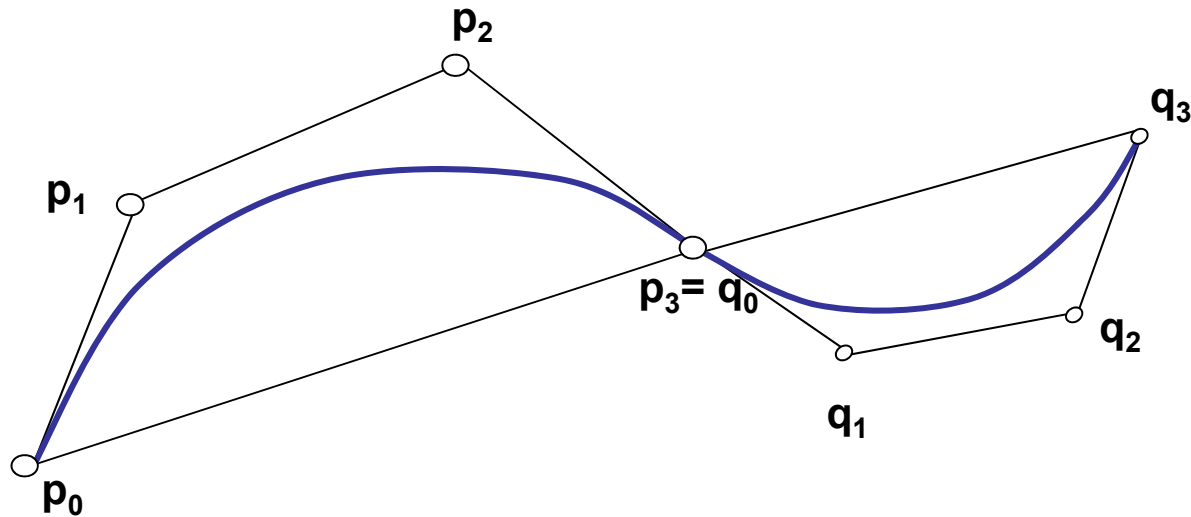
koncový bod = počáteční $p_3 = r_0$ a

shodné tangenty: $p_3 - p_2 = r_1 - r_0$ (bod $p_3 = r_0$ ve středu úsečky p_2r_1)

Napojování Bézierových oblouků - spojitost G^1



Spojitost G^1 (geometrická)



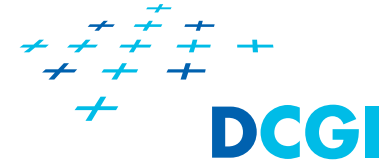
Spojitost G^1

koncový bod = počáteční $p_3 = q_0$ a

shodné směry tangent: $p_3 - p_2 = k(q_1 - q_0)$, $k > 0$

(bod $p_3 = q_0$ leží na úsečce p_2q_1)

Coonsovy kubiky

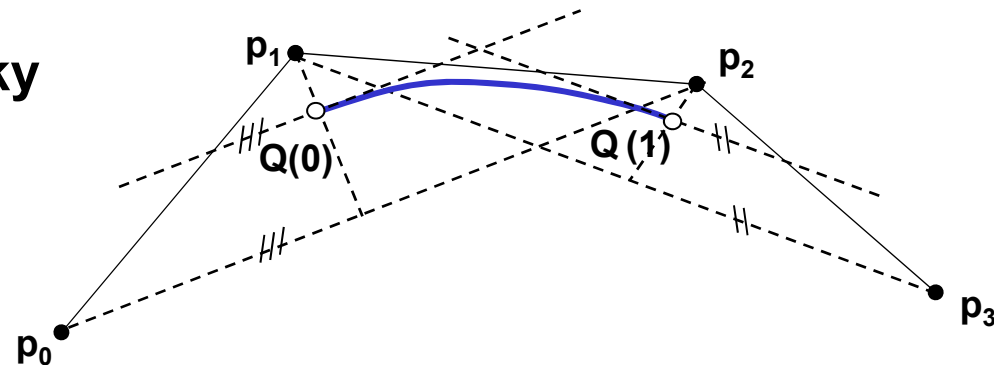


- Jiná forma interpretace kubik
- **Křivka vůbec neprochází uzlovými body, ale je přitahována k těmto bodům a to ke všem**
- Pozice sousedních bodů definuje tečnu v daném bodě

Vlastnosti Coonsovy kubiky



Vlastnosti Coonsovy kubiky



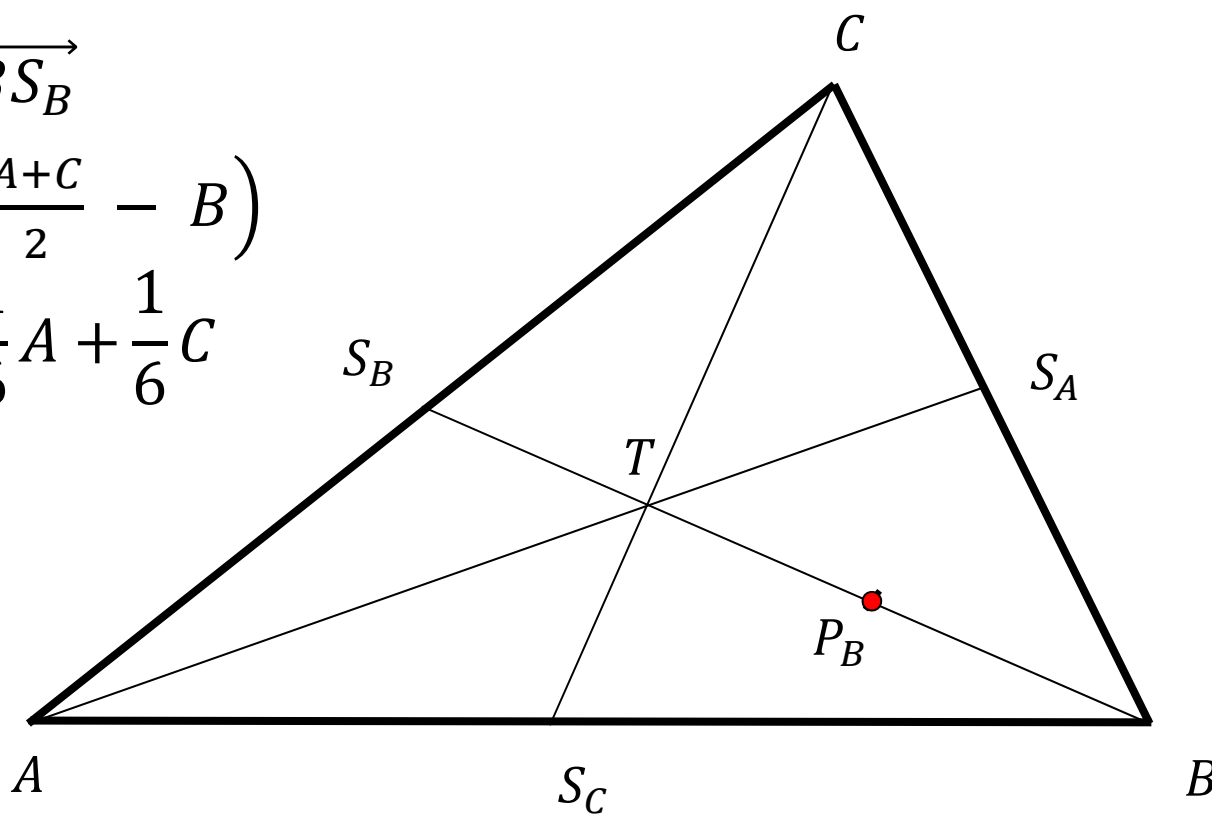
1. Křivka **začíná v antitěžišti** trojúhelníka p_0, p_1, p_2 ,
končí v antitěžišti trojúhelníka p_1, p_2, p_3
2. **Tečna** v bodě $Q(0)$ je **rovnoběžná s úsečkou p_0, p_2**
tečna v bodě $Q(1)$ je rovnoběžná s úsečkou p_1, p_3
tečný vektor má délku rovnou polovině úsečky p_1, p_3
3. Leží-li body p_0, p_1, p_2 , na přímce, leží bod $Q(0)$ na této přímce
a ta je současně tečnou oblouku v $Q(0)$. Analogicky pro p_1, p_2, p_3
4. $p_0 = p_1, \Rightarrow$ leží $Q(0)$ leží na přímce p_0, p_2 .
5. $p_0 = p_1 = p_2 \Rightarrow Q(0) = p_0$
 \Rightarrow Coonsova kubika degeneruje na úsečku (p_0, p_3)

Těžiště a antitěžiště trojúhelníku

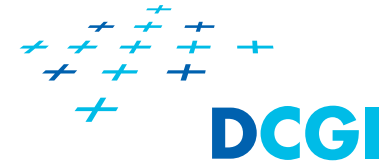


Antitěžiště P_B

$$\begin{aligned} P_B &= B + \frac{1}{3} \overrightarrow{BS_B} \\ &= B + \frac{1}{3} \left(\frac{A+C}{2} - B \right) \\ &= \frac{2}{3} B + \frac{1}{6} A + \frac{1}{6} C \end{aligned}$$



Coonsova kubika – segment spline-křivky



Řídící body p_i , $i = 0, 1, 2, 3$

$$Q(t) = \frac{1}{6} \sum_{i=0}^3 p_i \cdot C_i(t), \quad t \in 0..1,$$

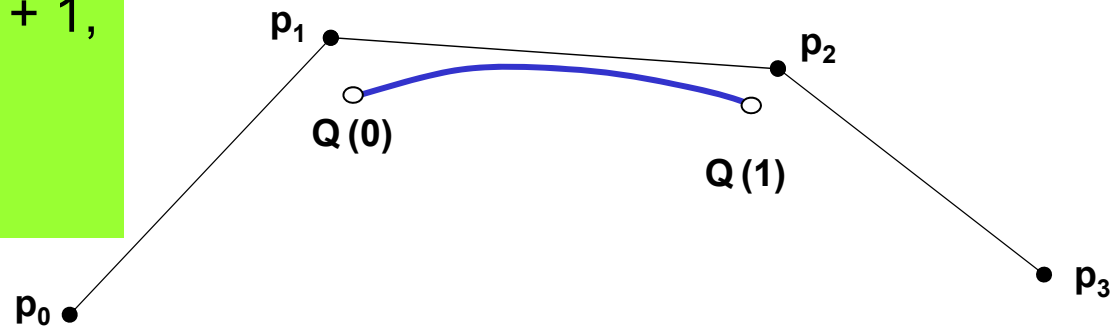
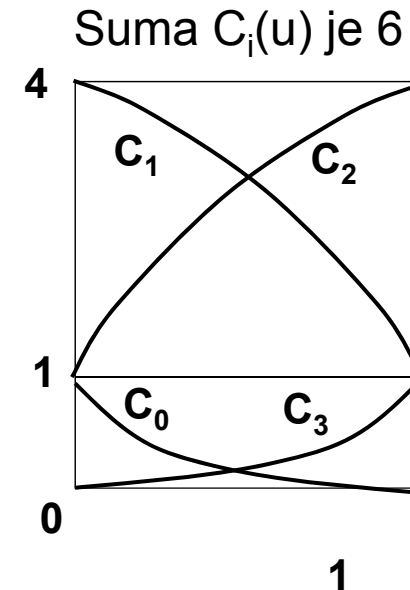
$$C_0(t) = (1 - t)^3,$$

$$C_1(t) = 3t^3 - 6t^2 + 4,$$

$$C_2(t) = -3t^3 + 3t^2 + 3t + 1,$$

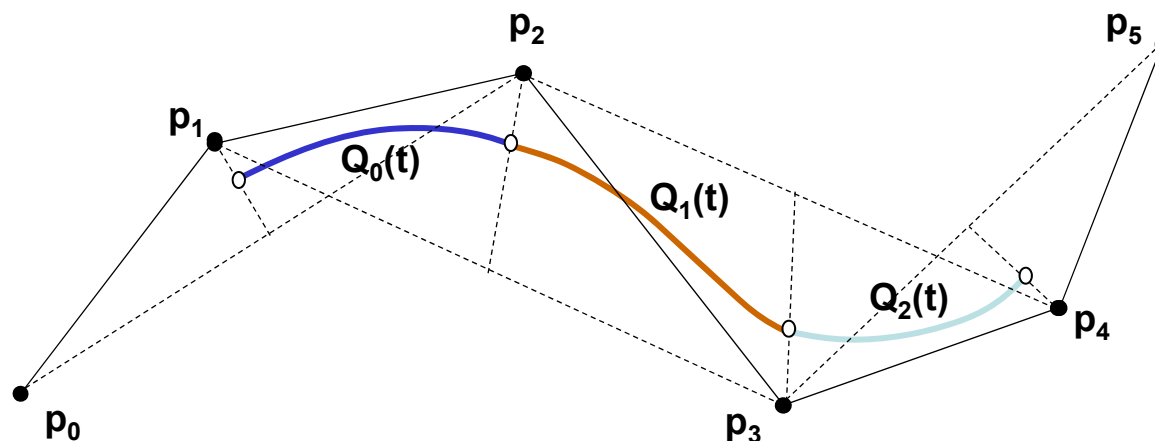
$$C_3(t) = t^3,$$

$$0 \leq t \leq 1$$



Napojování Coonsových oblouků

Napojování Coonsových oblouků



Stupeň napojení $C^2 \Rightarrow$ spojitě druhé derivace
spline křivka

Lokalita změn

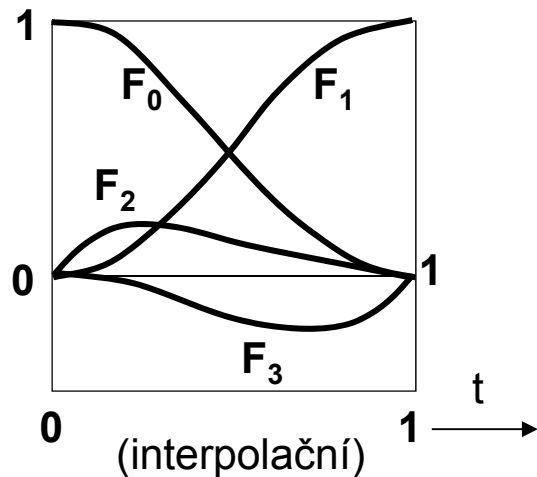
Velmi vhodná pro modelování

Báze kubických parametrických křivek

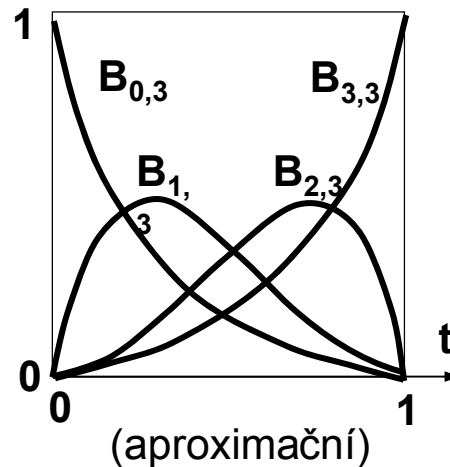


Ferguson

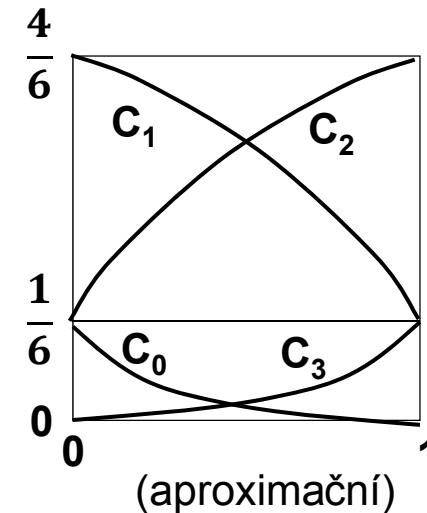
(hermitovské kubiky)



Bézier



Coons



Vždy 4 báze, funkce na intervalu 0 až 1,
křivka je dána superpozicí základních bází s různými koeficienty =
řídícími či uzlovými body

Plochy



Fergusonův plát

Vstup

1. Okrajové křivky plátu $p_0(v) = P(0,v)$, $p_1(v) = P(1,v)$.
2. Průběhy příčných tečných vektorů
 $t_0(v) = (\delta P(u,v)/\delta u)_{u=0}$, $t_1(v) = (\delta P(u,v)/\delta u)_{u=1}$
podél okrajových křivek $p_0(v)$ a $p_1(v)$

Vektorová rovnice plátu

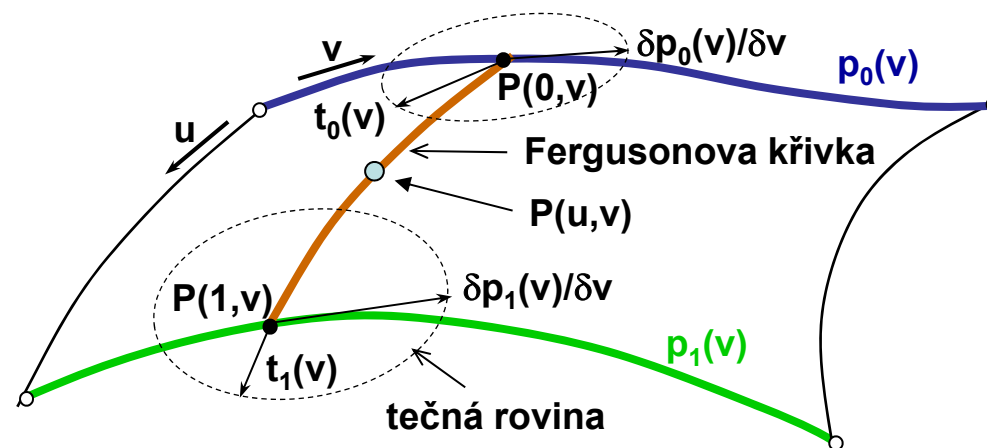
$$P(u,v) = F_0(u) \cdot p_0(v) + F_1(u) \cdot p_1(v) + F_2(u) \cdot t_0(v) + F_3(u) \cdot t_1(v),$$

$$F_0(u) = 2u^3 - 3u^2 + 1,$$

$$F_1(u) = -2u^3 + 3u^2,$$

$$F_2(u) = u^3 - 2u^2 + u,$$

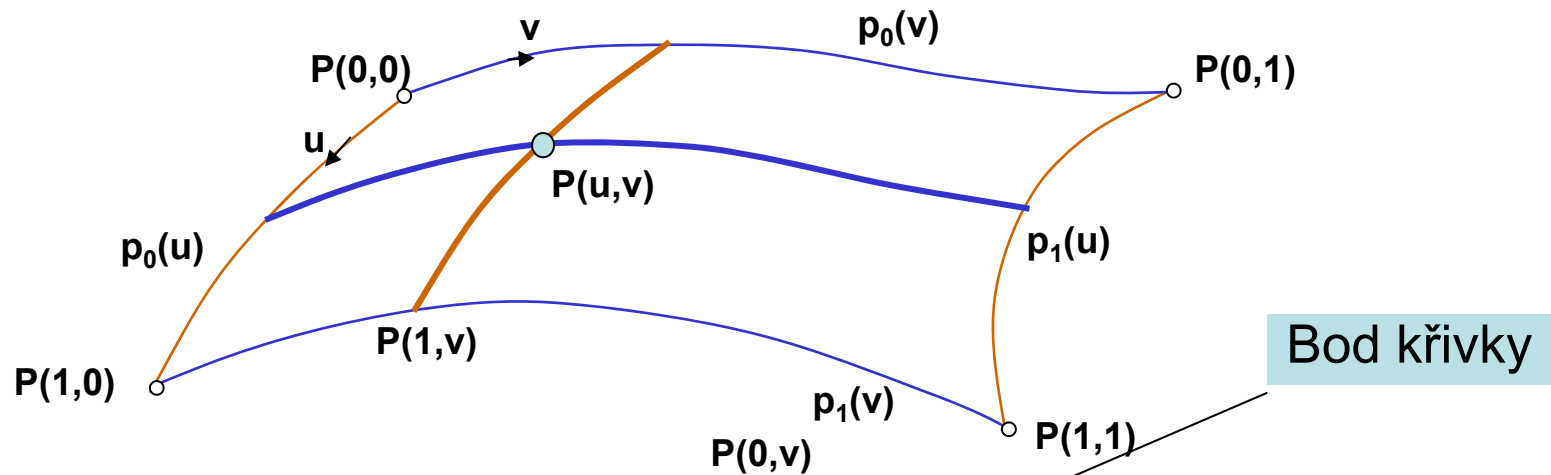
$$F_3(u) = u^3 - u^2.$$



Interpolační metody modelování ploch



Coonsova bikubická interpolační plocha (zadaná implicitně)



$$[F_1(u), -1, F_2(u)] \begin{bmatrix} P(0,0) & p_0(v) & P(0,1) \\ p_0(u) & P(u,v) & p_1(u) \\ P(1,0) & p_1(v) & P(1,1) \end{bmatrix} \begin{bmatrix} F_1(v) \\ -1 \\ F_2(v) \end{bmatrix} = 0$$

$$F_1(t) = 2 t^3 - 3 t^2 + 1$$

$$F_2(t) = -2 t^3 + 3 t^2$$

Bézierova dvouparametrická plocha



Bézierovy plochy

Dvouparametrická plocha

$$P(u, v) = \sum_{i=0}^n \sum_{j=0}^m p_{i,j} \cdot B_{i,n}(u) \cdot B_{j,m}(v), \quad u, v \in \langle 0, 1 \rangle,$$

$p_{i,j}$, $i = 0, 1, 2, \dots, n$, $j = 0, 1, 2, \dots, m$ - soustava řídicích bodů

$B_{i,n}(u)$, resp. $B_{j,m}(v)$ - Bernsteinovy polynomy

Vlastnosti Bézierových ploch

1. Rohové body sítě jsou rohovými body plochy.
2. Okrajové křivky plochy jsou Bézierovy křivky určené krajními stranami sítě.
3. Rohové strany sítě jsou tečnami okrajových křivek plochy
4. $P(u,v)$ se posune ve směru posunu kteréhokoliv bodu $p_{i,j}$

Bézierovy křivky a plochy v OpenGL



Bézierovy křivky v OpenGL

Postup - Křivka je dána sekvencí úseček generované pomocí jednoho cyklu nad parametrem

Bézierovy plochy v OpenGL

Postup - Plochu popíšeme dvojrozměrně parametricky pomocí dvou vnořených cyklů a vytváříme nejčastěji triangle strip. Normály či parametrizaci pro textury ukládáme spolu s generovanými vrcholy.

Adaptivní řešení pomocí teselačních shaderů

Reference



[Demidov] <http://www.ibiblio.org/e-notes/Splines/Intro.htm>,

updated 12 Feb 2004, stav dne 20.5.2006.

[Alexandr] Lubomír Alexandr. Výuka počítačové grafiky cestou WWW.
Diplomová práce. <http://www.hyperkrychle.cz/curves/obsah.html>

[MPG] Žára, Beneš, Sochor, Felkel. Moderní počítačová grafika, 2.
vydání, Computer Press, 2004, kap. 5.