KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

Křivky

Počítačová grafika

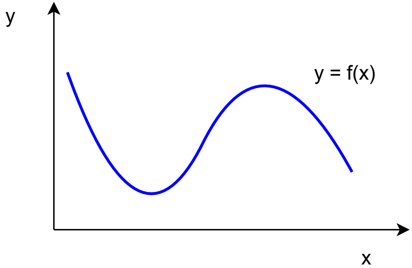
Mgr. Markéta Trnečková, Ph.D.

Zadání křivky

- **Zadání**:
  - Explicitní
  - Implicitní
  - Parametrické

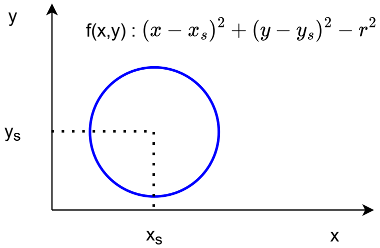
Explicitní křivky

- $y = f(x)$



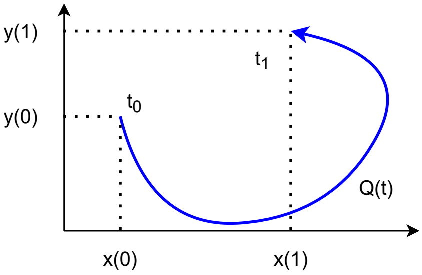
Implicitní křivky

- $f(x, y) = 0$
- **Příklad**: kružnice
  $(x - x_s)^2 + (y - y_s)^2 - r^2 = 0$
- obtížně zobrazitelné



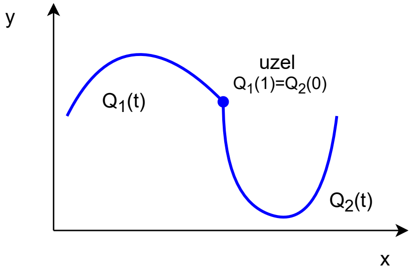
Parametrické zadání

- Dráha bodu v čase $t \in < t_{min}, t_{max} >$
- $x = x(t)$, $y = y(t)$
- **Bodová rovnice**: $Q(t) = [x(t), y(t)]$
- **Vektorová rovnice**: $\overrightarrow{q}(t) = [x(t), y(t)]$
- **Polohový vektor** $\overrightarrow{q}(t) = Q(t) - [0, 0]$
- Jednoduché zobrazení – závisí jen na jednom parametru t



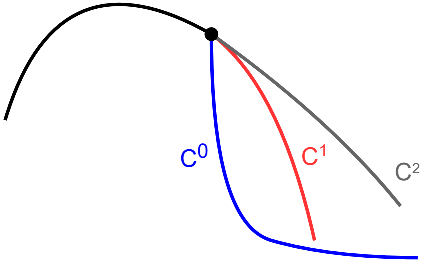
Složitější křivky

- Složitější křivky dělíme na **segmenty**
- je nutné řešit návaznost – **uzel**
- **Tečný vektor** $\overrightarrow{q'}(t_0) = (x'(t_0), y'(t_0))$
- **Rovnice tečny**
  $P(u) = Q(t_0) + u\overrightarrow{q'}(t_0) = (x'(t_0), y'(t_0))$
- **směrový vektor přímky** $\vec{q}'(t_0)$
- **Spojitost**:
  - Parametrická
  - Geometrická
- **Třída křivky**



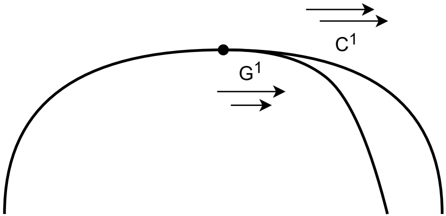
Parametrická spojitost

- Parametrické spojitost stupně n
- **Označení**: C^n
- Ve všech bodech má spojité derivace podle parametru t do řádu n
- C^0 = segmenty jsou spojitě navázány
- C^1 = tečný vektor v koncovém bodě segmentu Q_1 je roven tečnému vektoru segmentu Q_2
- $\vec{q_1}^{(i)}(1) = \vec{q_2}^{(i)}(0), \forall i = 0, \ldots, n$



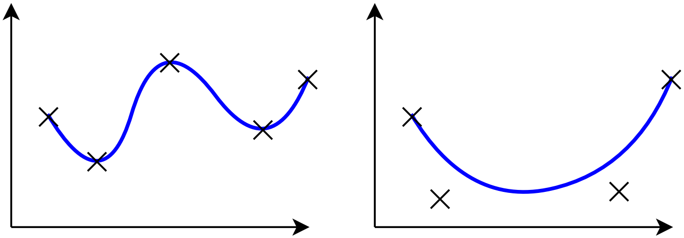
Geometrická spojitost

- Geometrická spojitost stupně n
- **Označení**: G^n
- Ve všech bodech má spojité derivace podle parametru t do řádu n
- C^0 = koncový bod prvního segmentu je počátečním bodem druhého
- C^1 = tečné vektory $\vec{q_1}(1)$ a $\vec{q_2}(0)$ jsou souhlasně kolineární
- $\vec{q_1}(1) = k \cdot \vec{q_2}(0)$, pro $k > 0$



Modelování křivek

- **Polynomiální křivky**
- $Q_n(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n$
- Modelují se pomocí **řídících bodů**
- Řídící body tvoří **řídící polynom**
- Řídící body se:
  - Interpolují
  - Aproximují



Kubiky

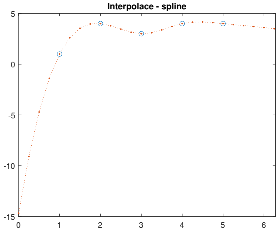
- Polynomiální křivky 3. stupně
- Můžeme zaručit spojitost C^2
- **Parametrické zadání**:
  $x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$
  $y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$
- **Maticové zadaní**:
  $$Q(t) = TC = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix}$$
- Závisí na 8 parametrech

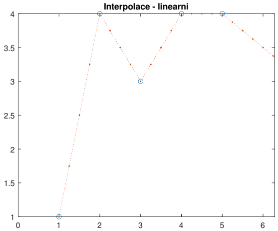
Obecné vlastnosti křivek

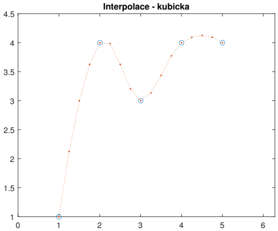
- Křivka má procházet krajními body.
- Pokud na křivku aplikujeme nějakou lineární transformaci (případně projekci), pak bychom měli dostat stejný výsledek, jako bychom tuto transformaci aplikovali na řídící body a pak křivku podle nich vykreslili.
- Vlastnosti konvexní obálky:
  - silná podmínka: celá křivka leží v konvexní obálce svých řídících bodů.
  - slabá podmínka: část křivky leží v konvexní obálce některých řídících bodů (segment leží v obálce svého řídícího polynomu).
- Změnou polohy a váhy řídícího bodu se mění jen část křivky, nikoliv křivka celá.

Interpolační křivky

- **Matlab**: `vystup_y = interp1(vstup_x, vstup_y, vystup_x, metoda);`

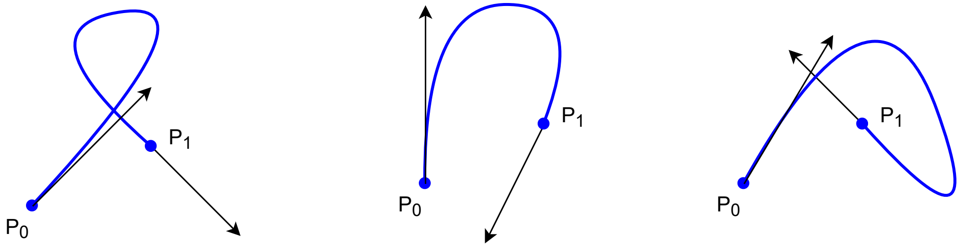






Hermitovské kubiky

- Interpolační křivky
- **Řídící body**: P_0 a P_1
- **Tečné vektory koncových bodů**: $\vec{p_0}'$ a $\vec{p_1}'$
- $P(t) = (2t^3 - 3t^2 + 1)P_0 + (t^3 - 2t^2 + t)\vec{p_0}' + (-2t^3 + 3t^2)P_1 + (t^3 - t^2)\vec{p_1}'$



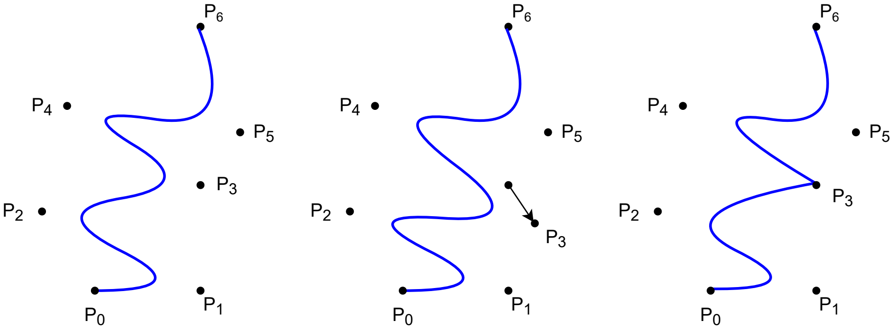
Aproximační křivky

- Křivka nemusí procházet body, jen se k nim blížit
- **Příklad**:
  - Beziérovy křivky
  - Coonsovy křivky
  - Spline křivky

Beziérovy křivky

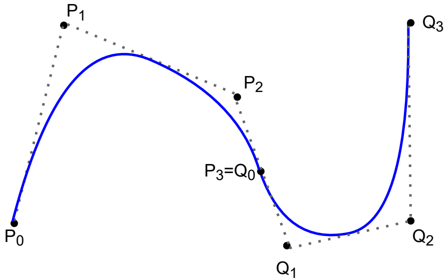
- **Použití** – definice fontů
- Křivka prochází prvním a posledním bodem a leží v konvexní obálce řídících bodů
- Invariantní vůči posunu, změně měřítka a otočení
- Změna polohy bodu má vliv na tvar celé křivky
- **Beziérova křivka stupně** n: $Q(t) = \sum_{i=1}^{n} P_i B_i^n(t)$
- B_i jsou *Bernsteinovy polynomy n-tého stupně*
- $B_i^n(t) = \binom{n}{i} t^i (1 - t^{n-i})$
- Pro $t \in < 0, 1 >$, $i = 0, \ldots, n$

Beziérovy křivky



Beziérovy křivky

- Složitější křivky dělíme nejčastěji na kubiky
- **Spojitost** C^0: snadná
- **Spojitost** C^1: bod $P_n = Q_0$ středem úsečky určené body P_{n-1} a Q_1
- $\vec{q_0}' = n(Q_1 - Q_0)$
  $\vec{p_1}' = n(P_n - P_{n-1})$
- **Spojitost** G^1: P_{n-1}, $P_n = Q_0$ a Q_1 leží na jedné přímce



Vykreslení Beziérových křivek v rastru

- **Algoritmy**:
  - naivní (neadaptivní)
  - rekurzivní algoritmus de Castlejau

Naivní algoritmus

- **Zadání křivky**: $Q(t) = \sum_{i=1}^{n} P_i B_i^n(t)$
- Postupně dosazujeme t a body spojíme úsečkami
- Δt – konstantní (nestejně dlouhé úseky křivky)

Algoritmus de Castlejau

- Výpočet bodu $Q(t) = P_{n,n}$
- $P_{j,i}(t) = (1-t)P_{j-1,i-1} + tP_{j,i-1}$
- $i = 1, \ldots, n$ a $j = i, i+1, \ldots, n$
- **Vstup**: body řídícího polygonu ($P_{i,0} = P_i$)
- **Postup výpočtu** $Q(1/2)$ ($P_{3,3}$)
- Každé rozdělení generuje dva nové řídící polygony