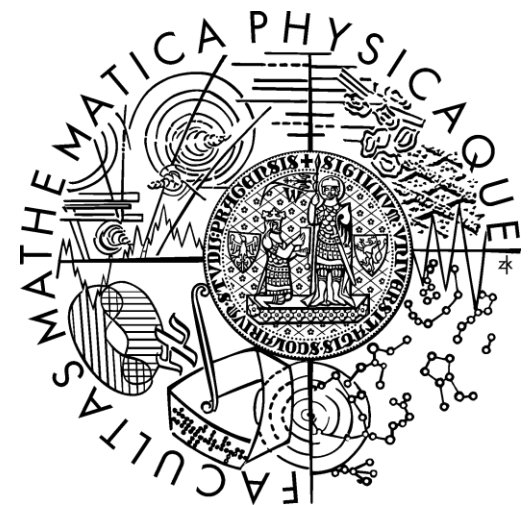


# Rekonstrukce křivek a ploch metodou postupné evoluce

**Petra Surynková**

Matematicko-fyzikální fakulta  
Univerzita Karlova v Praze

[petra.surynkova@mff.cuni.cz](mailto:petra.surynkova@mff.cuni.cz)



# Přehled (1)

- **Princip rekonstrukce ploch technické praxe z mračna bodů**
  - Bodová fáze – příprava a předzpracování
  - Polygonální reprezentace
  - Tvarová fáze – segmentace, aproximace oblasti daným typem plochy
- **Hraniční reprezentace**
  - Polygonální reprezentace – příklady v programu Rhinoceros
  - Analytická reprezentace
- **Analytická vyjádření hraničně reprezentovaných objektů**
  - Bézierovy plochy
  - Racionální Bézierovy plochy
  - Plátování
  - Příklady ploch



# Přehled (2)

- **Aproximace ploch Bézierovými pláty**
  - Body vstupní množiny tvoří pravidelnou mřížku
  - Body vstupní množiny jsou rozmístěny náhodně
  - Programová demonstrace
- **Další vylepšování tvaru aproximující plochy**
  - Princip evoluce křivek
  - Evoluce uzavřených a otevřených B-spline křivek
  - Princip evoluce ploch
  - Evoluce Bézierových ploch
  - Programová demonstrace
- **Závěr a budoucí práce**



# Rekonstrukce ploch (1)

- **Digitální dokumentace reálných povrchů**
  - **rekonstrukce ploch** je využívána v řadě odvětvích vědy a průmyslu
    - dokumentace soch a památek, architektura, stavitelství, design, ...
  - **vytvoření CAD modelu** z daného mračna bodů
    - body v prostoru, v obecné úloze známe jen prostorové souřadnice
- **Bodová fáze – příprava a předzpracování**
  - mračna bodů získáváme **3D skenováním** reálných povrchů
    - nutné počítat s nepřesností v měření
    - **skenování** se provádí **vícekrát z různých stanovišť**, jednotlivé „skeny“ se následně slučují
    - ve výsledném mračnu bodů mohou být **redundantní data** – nutné odstranit – ztenčující algoritmy (thinning algorithms)



# Rekonstrukce ploch (2)

- **Polygonální reprezentace**

- celá řada algoritmů využívající dělení prostoru
  - základem Voronoiovy diagramy, Delaunay triangulace
- výsledkem těchto metod bývá velice často **trojúhelníková síť** aproximující zadanou množinu bodů
  - velice komplikovaný problém, dosud neexistují uspokojivé univerzální řešící postupy
  - př.:  $\alpha$  – shapes, crust algorithm, cocone algorithms

- **Tvarová fáze**

- převod trojúhelníkové sítě do CAD reprezentace
  - **implicitní nebo parametrické vyjádření ploch**
- **segmentace** – detekce a rozdělení na oblasti s „rozdílnou geometrií“
  - tj. nalezení **rovinné oblasti**, části **válcových ploch** nebo **obecné plochy** (freeform patches) – není jednoduché
- **aproximace oblasti typem plochy** určeného při segmentaci (surface fitting)

# Rekonstrukce ploch (3)



**Model projektu –  
Frank O. Gehry,  
Walt Disney  
Concert Hall**



**digitální  
rekonstrukce  
fyzického modelu  
přináší účinný  
prostředek v  
projektování**

**Walt Disney Concert Hall, Frank  
O. Gehry - Los Angeles, USA**









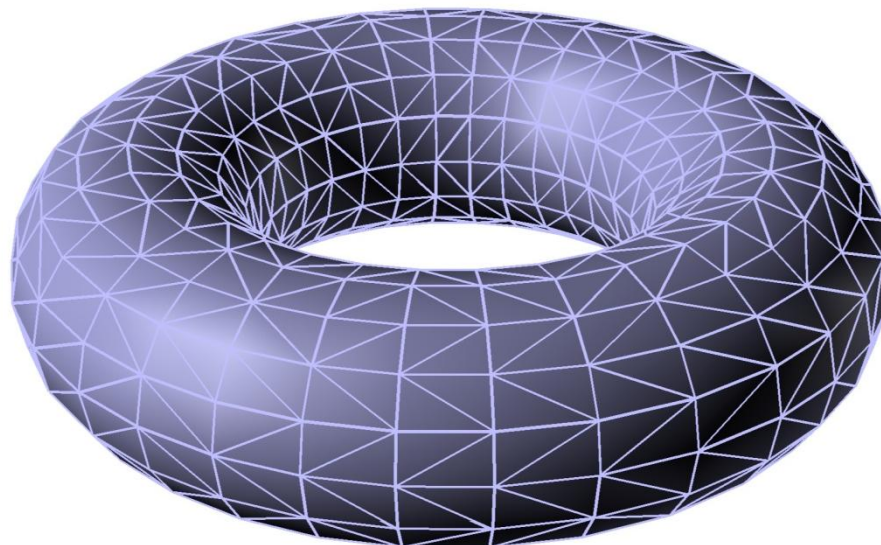
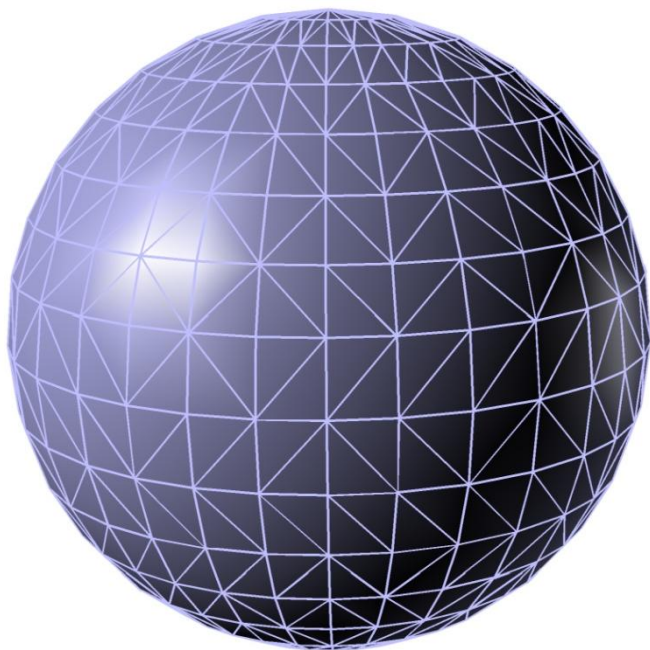
# Hraniční reprezentace (1)

- **Polygonální reprezentace**

- jedna z nejčastěji používaných reprezentací objektů v počítačové grafice
- základním prvkem
  - **nejčastěji trojúhelník**, někdy čtyřúhelník nebo mnohoúhelníky (nutné určovat konvexitu)
  - polygon je nejčastěji reprezentován **pouze svými vrcholy** – pořadí vrcholů a tudíž i pořadí jeho **stran** určuje **orientaci celého polygonu**
- při rekonstrukci ploch – **vstupem množina bodů**, známe **prostorové souřadnice**, obecně nevíme nic dalšího
  - díky polygonální reprezentaci, určíme **návaznost bodů vstupní množiny**
  - jde o vygenerování **prvotního povrchu plochy**, který se dále zpracovává

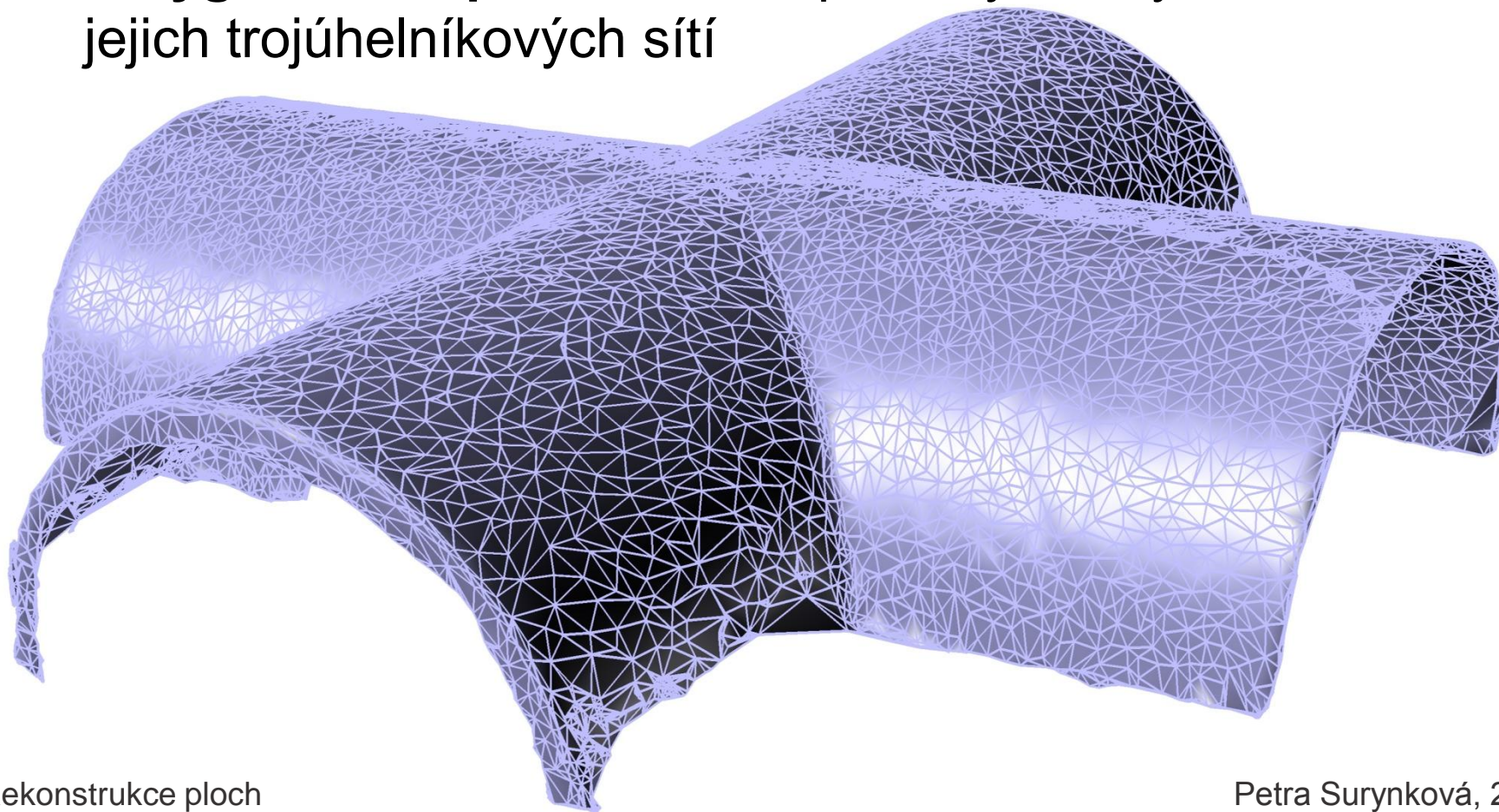
# Hraniční reprezentace (2)

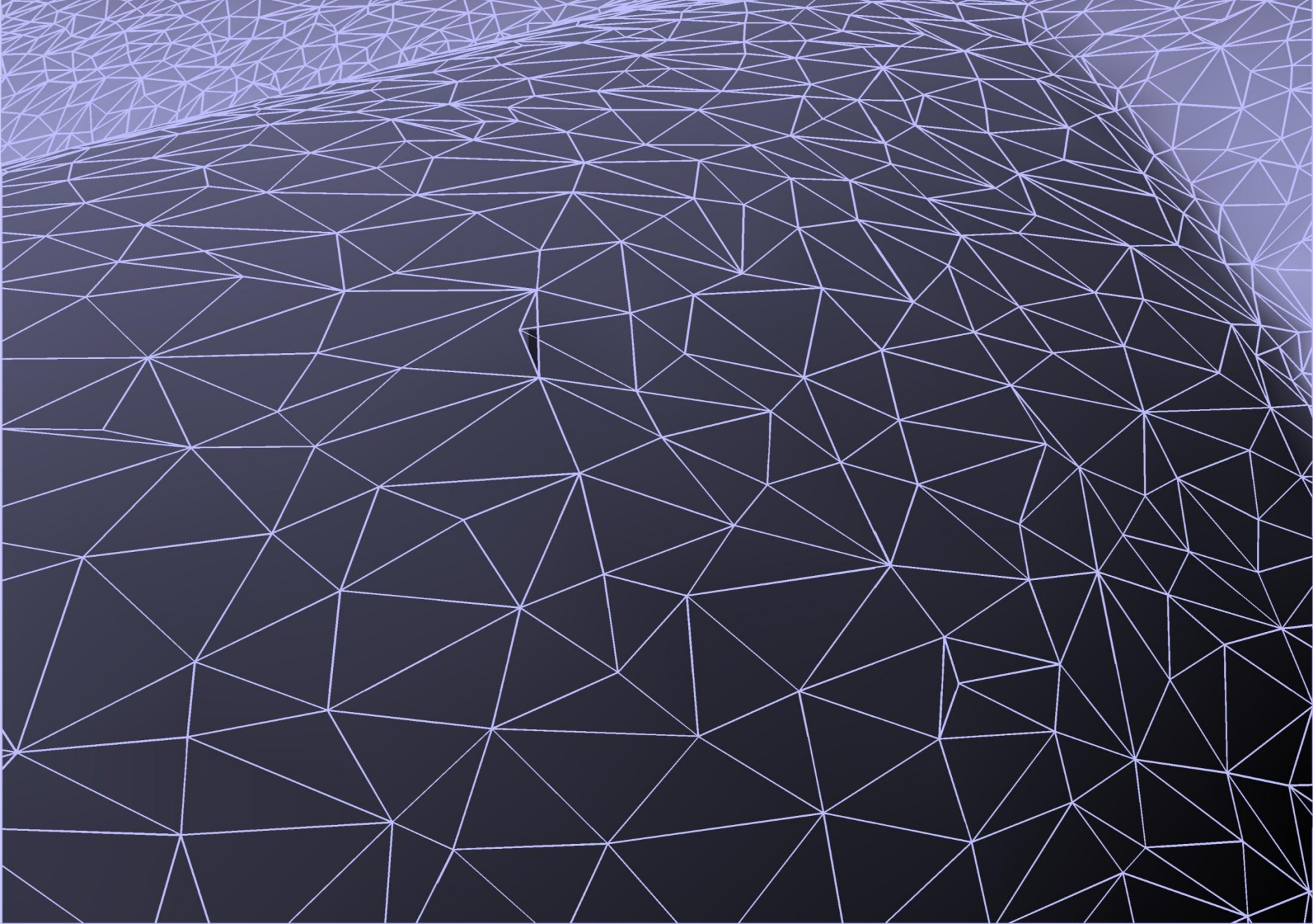
- **Polygonální reprezentace** – příklady trojúhelníkové sítě

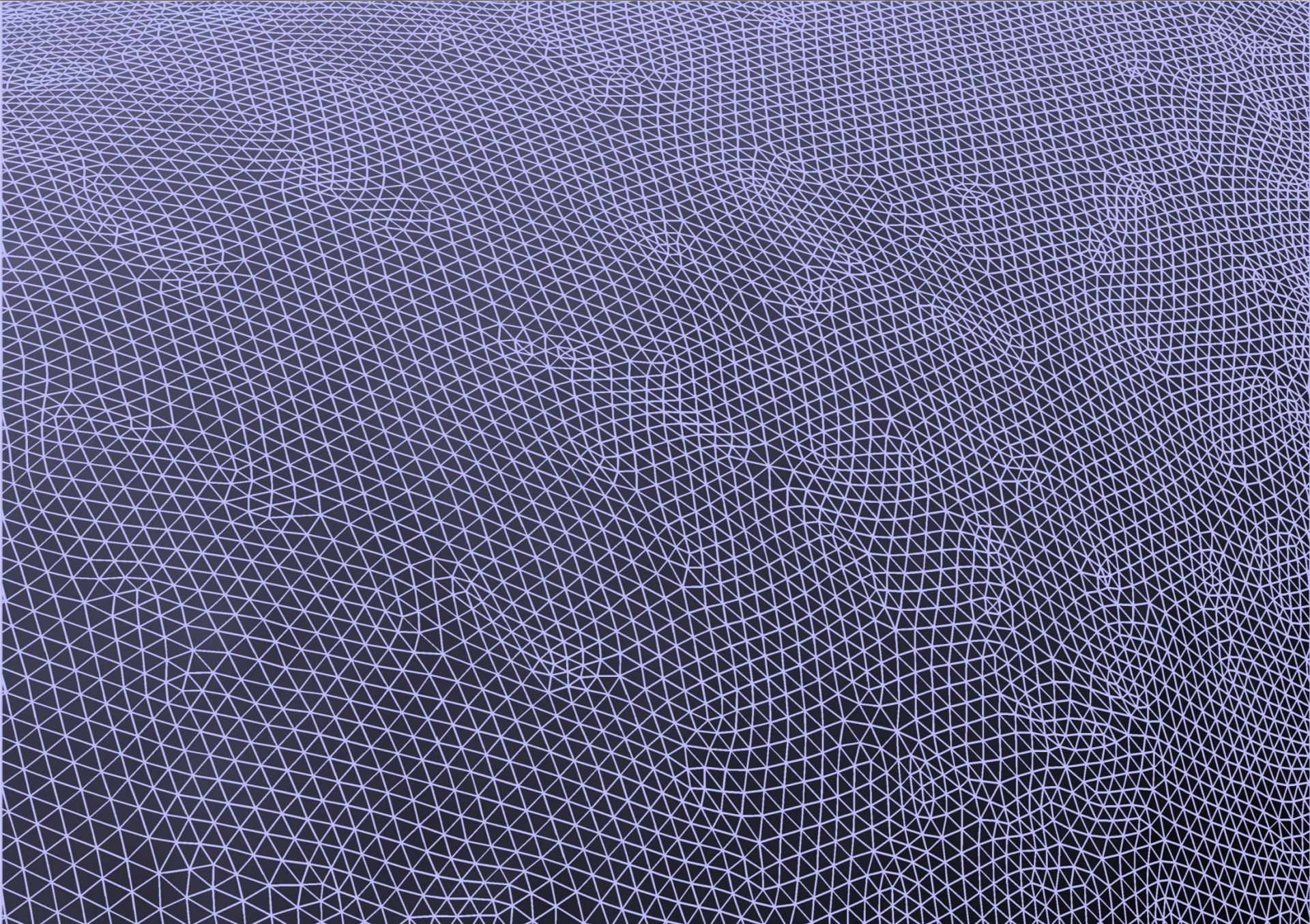


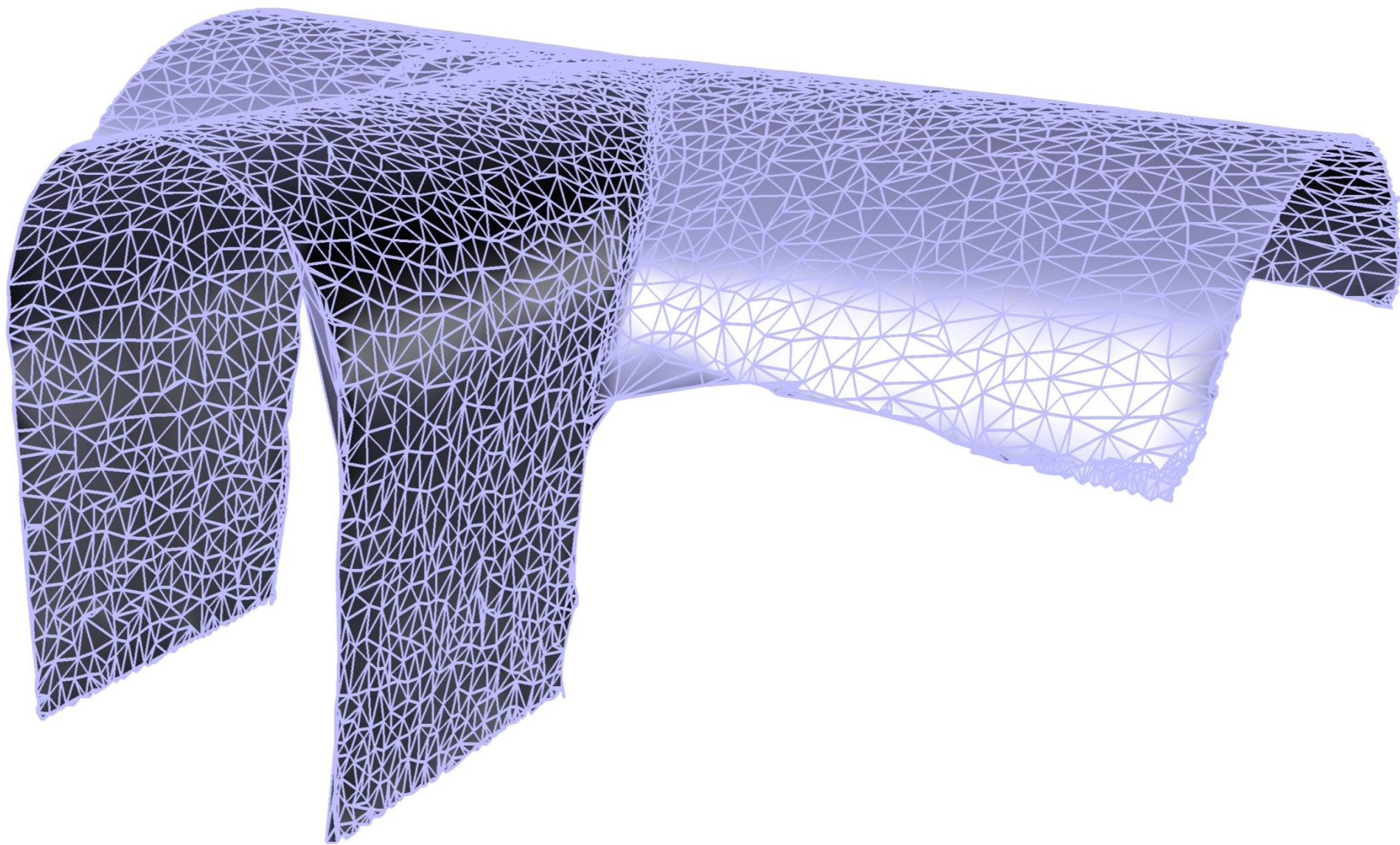
# Hraniční reprezentace (3)

- **Polygonální reprezentace** – příklady reálných dat a jejich trojúhelníkových sítí









# Ukázka trojúhelníkových sítí naskenovaných reálných objektů

**Využití modelovacích funkcí a nástrojů v  
programu Rhinoceros**



# Hraniční reprezentace (4)

- **Analytická reprezentace**

- rozsáhlá teorie diferenciální geometrie křivek a ploch
- nejčastěji parametrický popis, implicitní popis (méně)

- **výhody**

- základní předností analytického vyjádření ploch je jeho **přesnost**
  - například v architektuře vysoce žádoucí
- analytické vyjádření ploch umožňuje **měřit povrch**
- plochy lze poměrně **snadno editovat**
- lze **měnit tvar plochy**
- s plochami lze provádět různé operace
  - **dělení** nebo **průniky**
- parametricky vyjádřené funkce jsou **jednoduše diferencovatelné**
  - objekty takto popsané se snadno navazují

# Hraniční reprezentace (5)

- v analytické formě bývá obtížné (někdy dokonce nemožné) vyjádřit **složitý objekt jako celek**
- složité objekty proto popisujeme parametricky jako objekty složené z jednodušších částí
  - vytváření výsledných objektů napojováním ploch tzv. plátů se v počítačové grafice označuje jako **plátování**
- k popisu ploch technické praxe budeme používat **Bézierovy a racionální Bézierovy plochy**
  - definice, speciální příklady, které lze v rekonstrukci povrchů využít
  - napojování Bézierových plátů

# Bézierovy plochy (1)

- **Bézierova plocha stupně  $m \times n$** 
  - je určena  $(m+1) \times (n+1)$  řídicími body a vztahem

$$Q(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_i^m(u) B_j^n(v), u \in \langle 0, 1 \rangle, v \in \langle 0, 1 \rangle$$

- **bázové funkce  $B_i^k(t), k = n, m$  jsou Bernsteinovy polynomy  $k$ -tého stupně**

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

$$t \in \langle 0, 1 \rangle, i = 0, 1, \dots, k$$

# Bézierovy plochy (2)

- **Bézierova plocha stupně  $m \times n$**
- Bézierova plocha prochází rohovými body sítě a okrajové křivky plochy jsou Bézierovými křivkami pro okraje sítě
  - **tečná rovina v bodě  $P_{00}$  je určena body  $P_{00}, P_{10}, P_{01}$** 
    - **podobně pro další rohové body**

$$Q(0,0) = P_{00}$$

$$Q(0,1) = P_{0n}$$

$$Q(1,0) = P_{m0}$$

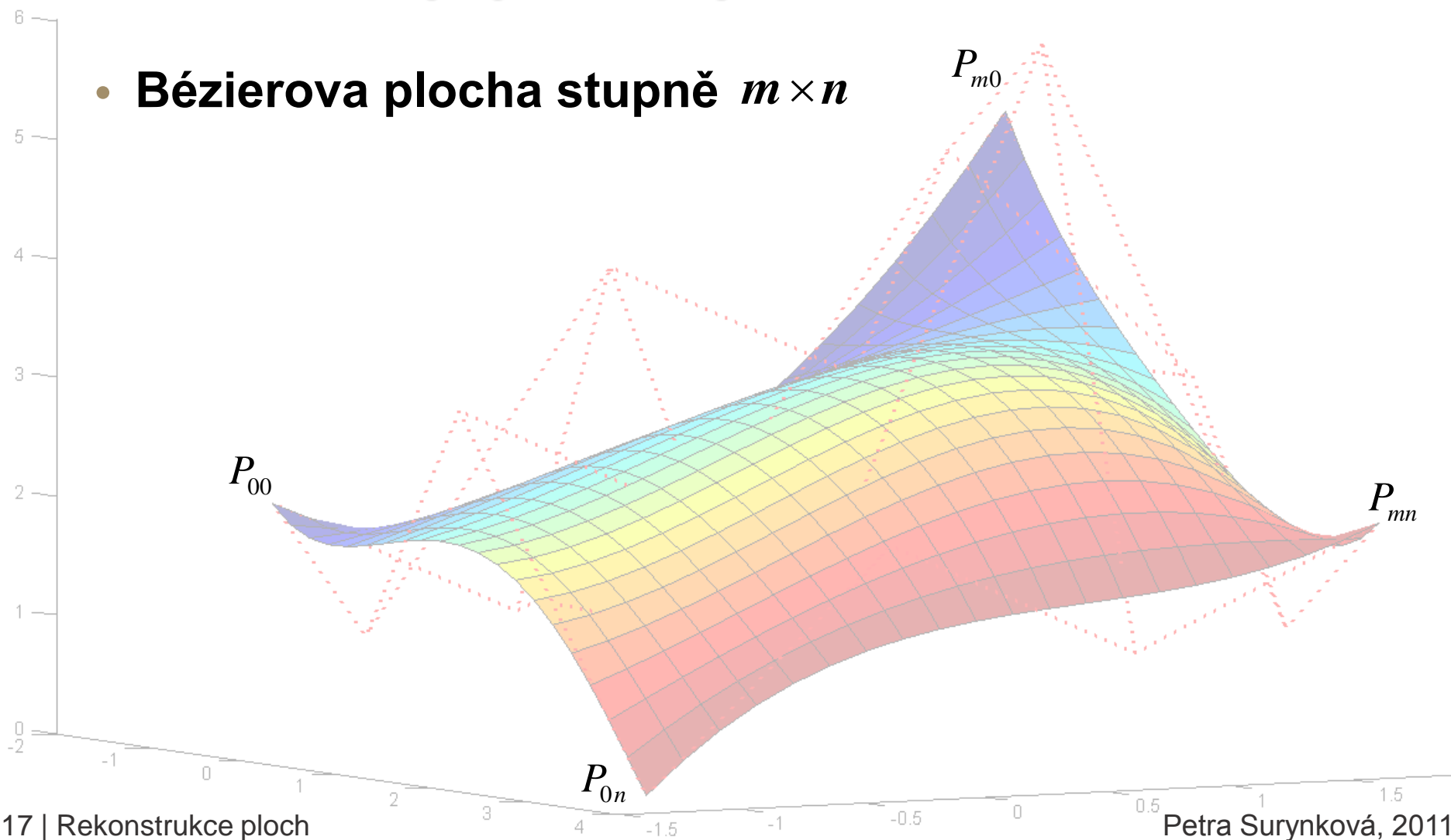
$$Q(1,1) = P_{mn}$$

**okrajová křivka např.:**

$$Q(0, v) = \sum_{j=0}^n P_{0j} B_j^n(v)$$

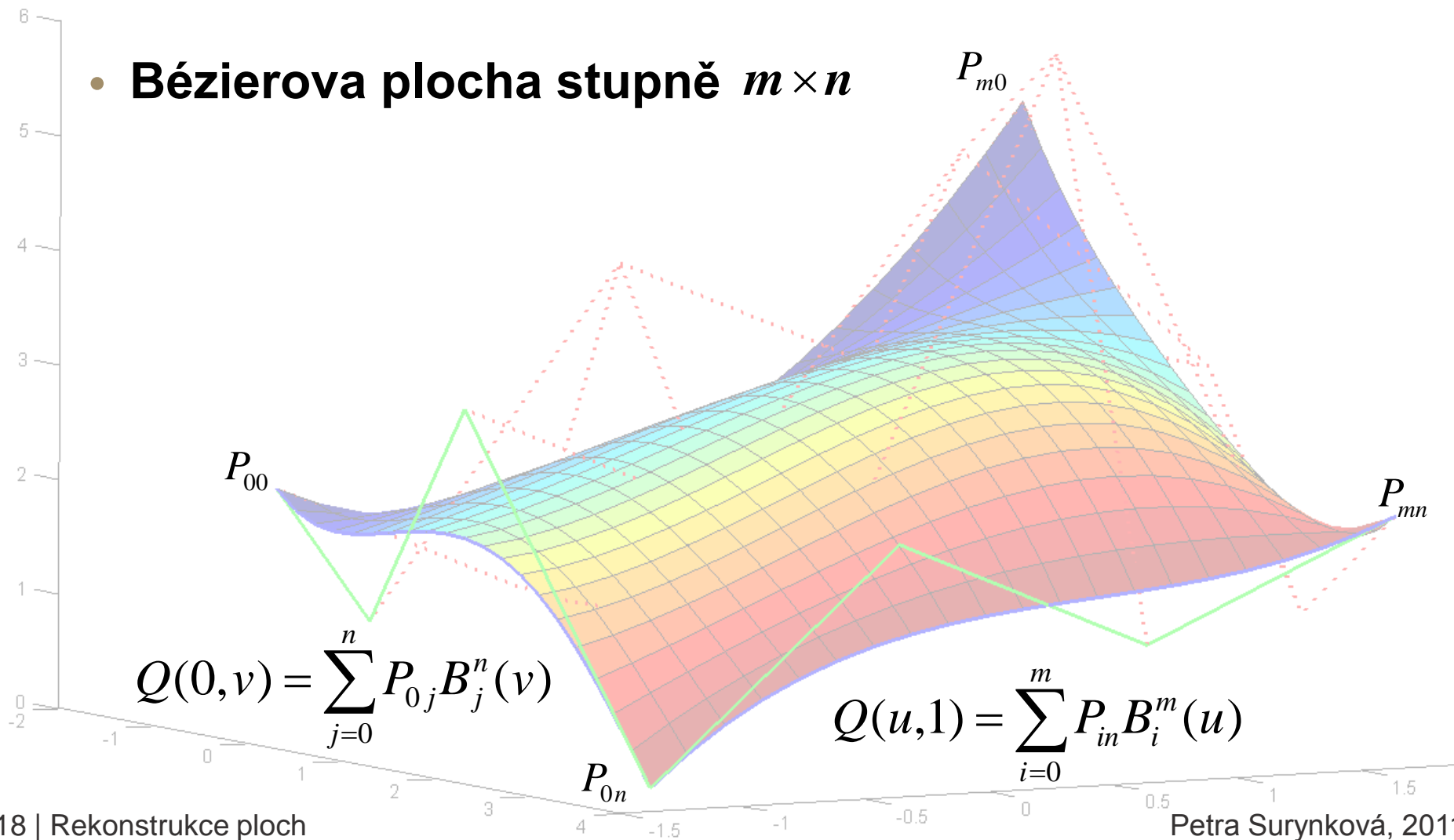
# Bézierovy plochy (3)

- Bézierova plocha stupně  $m \times n$



# Bézierovy plochy (4)

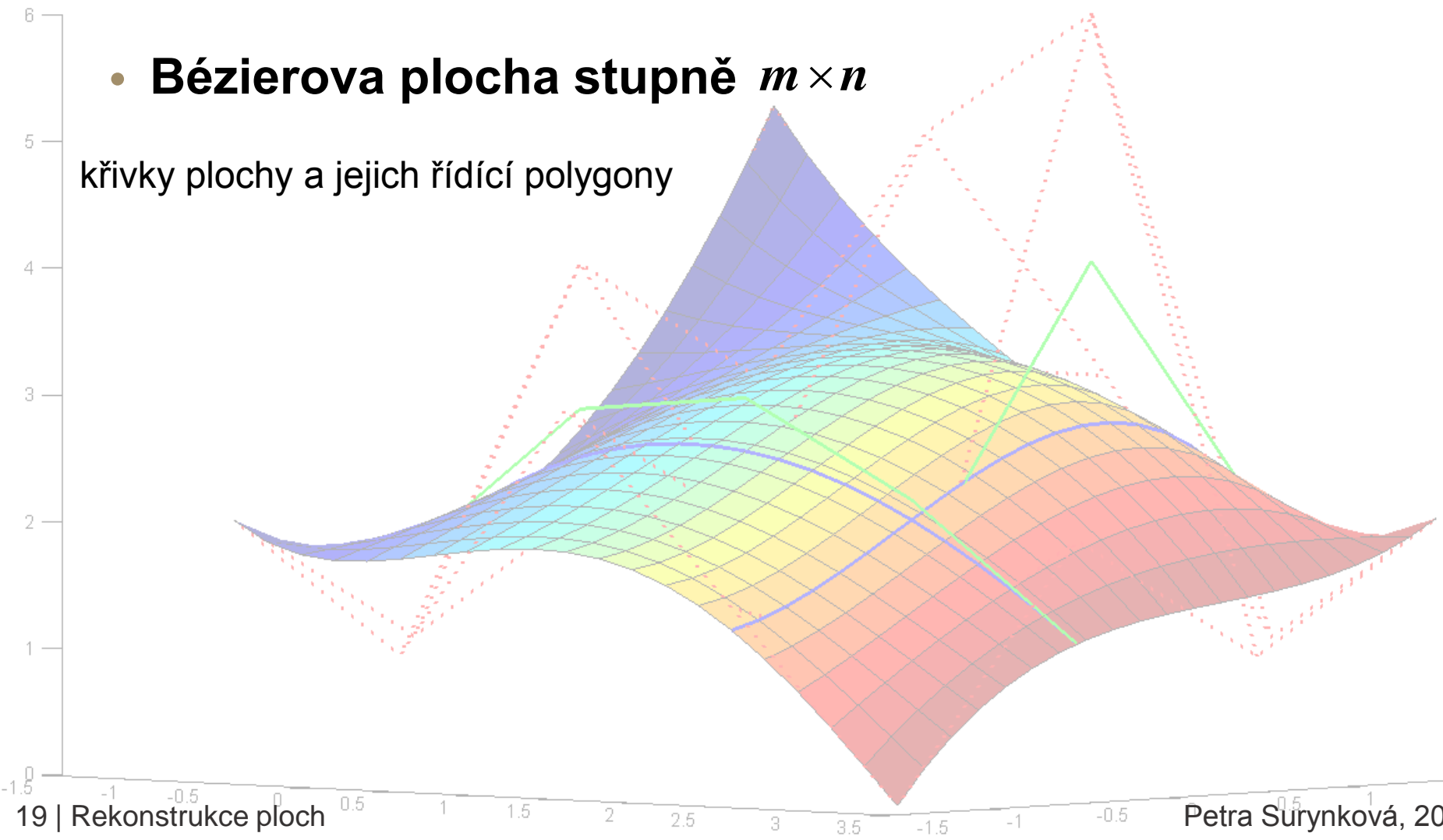
- Bézierova plocha stupně  $m \times n$



# Bézierovy plochy (5)

- **Bézierova plocha stupně  $m \times n$**

řivky plochy a jejich řídící polygony



# Bézierovy plochy (6)

- **Bézierova plocha stupně  $m \times n$**

křivky plochy a jejich řídící polygony





# Bézierovy plochy (7)

- **Bézierova bikubická plocha**  $m = 3, n = 3$

$$Q(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 P_{ij} B_i^3(u) B_j^3(v), u \in \langle 0, 1 \rangle, v \in \langle 0, 1 \rangle$$

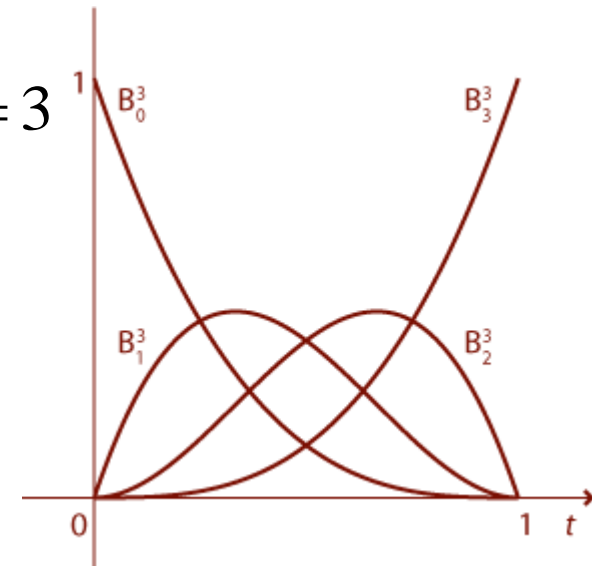
$$\begin{aligned}
 B_0^3(t) &= (1-t)^3 \\
 B_1^3(t) &= 3t(1-t)^2 \\
 B_2^3(t) &= 3t^2(1-t) \\
 B_3^3(t) &= t^3
 \end{aligned}
 \quad
 M_B = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}
 \quad
 P = \begin{pmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{pmatrix}$$

$$Q(u, v) = U M_B P M_B^T V^T = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} M_B P M_B^T \begin{pmatrix} v^3 \\ v^2 \\ v \\ 1 \end{pmatrix}$$

# Bézierovy plochy (8)

- **Bézierova bikubická plocha  $m = 3, n = 3$**

- v Matlabu nepočítáme se symbolickými proměnnými
- plně využíváme podporu počítání s maticemi
- určujeme **matici kořenů Bernsteinových polynomů**
- určujeme **matici koeficientů Bernsteinových polynomů  $M_B$** 
  - z kořenů Bernsteinových polynomů umíme zpětně určit koeficienty polynomů
- dále vyhodnocujeme Bernsteinovy polynomy v hodnotách parametrů  $u$  a  $v$ 
  - $u, v$  reprezentovány jako vektory, vyhodnocení Bern. pol. ukládáme do matic



$$M_B = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

# Bézierovy plochy (9)

- **Plátování**

- mějme dva Bézierovy pláty  $Q, R$
- první z nich je určen sítí řídicích bodů  $Q_{ij}, i = 0, \dots, s; j = 0, \dots, n$
- druhý je určen sítí řídicích bodů  $R_{ij}, i = 0, \dots, t; j = 0, \dots, n$ 
  - počet bodů ve směru  $v$  je stejný pro oba pláty a je roven  $n$
- pláty navazujeme ve směru  $u$

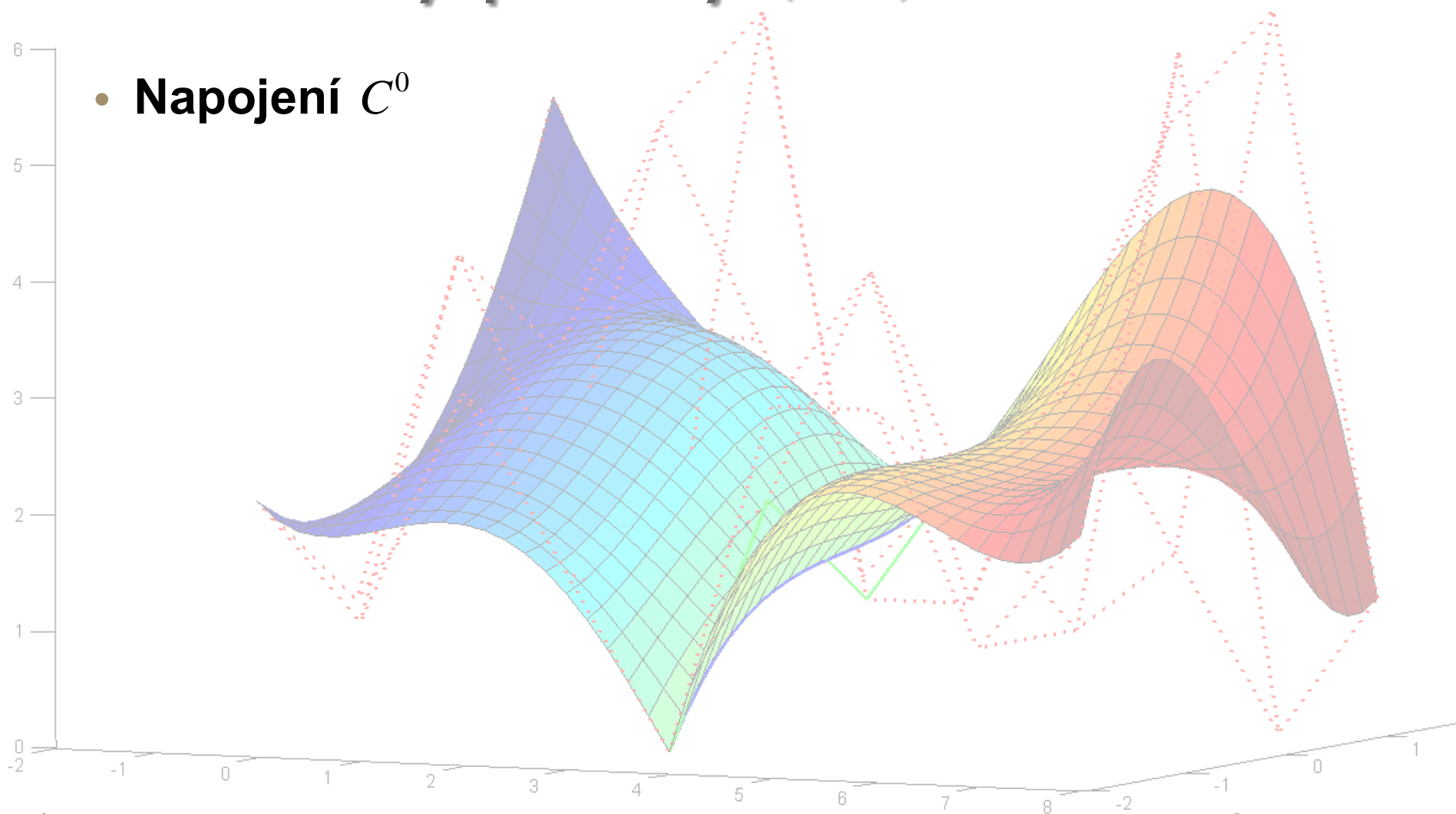
- **Napojení  $C^0$**

- **pláty mají společnou stranu**, tj.  $Q(u, 1) = R(0, v)$ , toho docílíme ztotožněním řídicích bodů, které určují příslušnou stranu
  - tj. pláty mají společnou hraniční lomenou čáru a různé příčné tečné vektory

$$Q_{sj} = R_{0j}, j = 0, \dots, n$$

# Bézierovy plochy (10)

- Napojení  $C^0$



# Bézierovy plochy (1 1)

- **Napojení  $C^1$** 
  - **pokud společná strana plátů je  $C^1$  spojitá a jsou-li identické příčné tečné vektory** ve směru  $u$  podél této strany
  - **tj. shoda  $C^1$  spojitě strany a splnění následující vztahu** pro body řídicích polygonů obou plátů

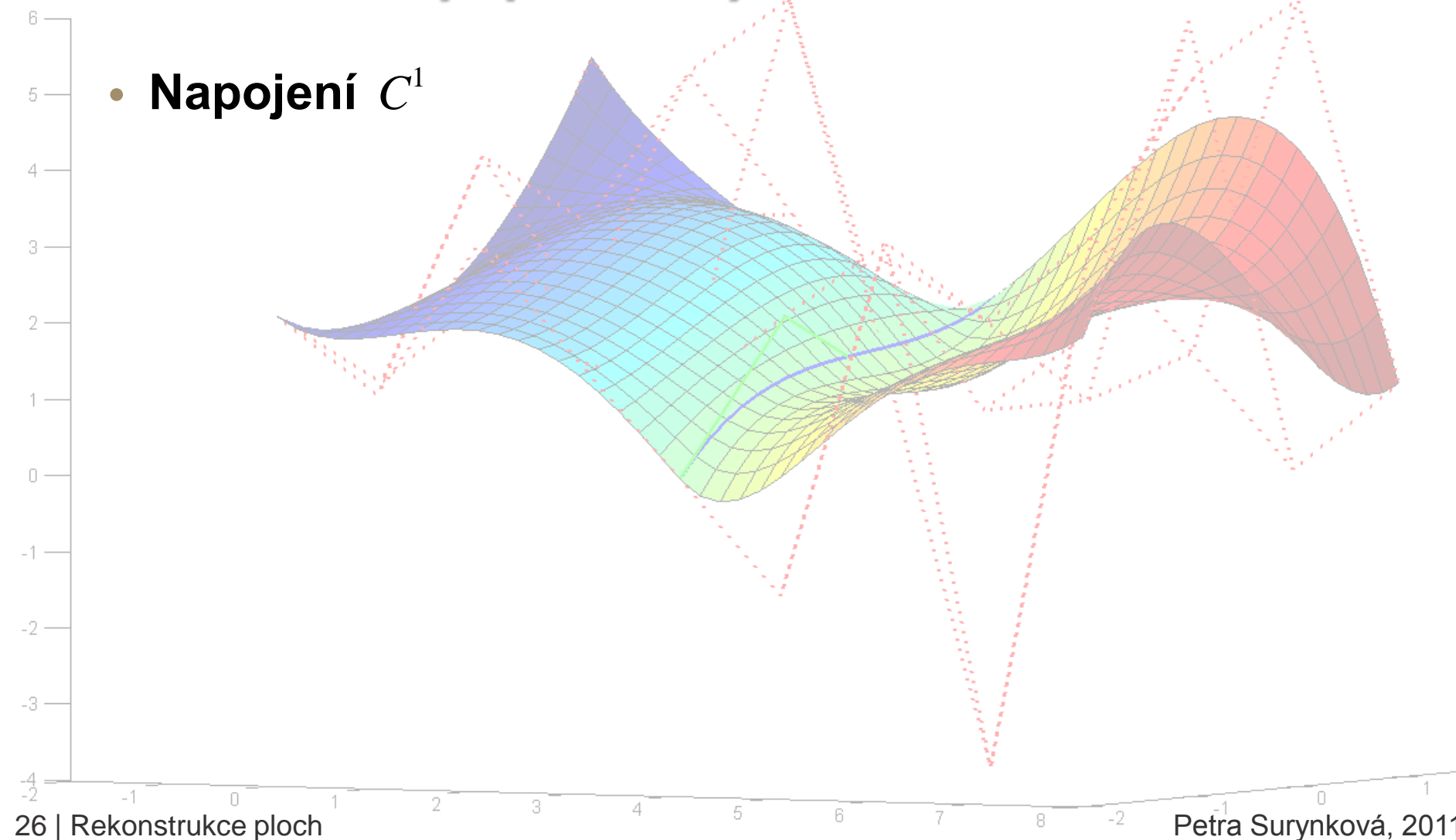
$$Q_{sj} - Q_{s-1j} = R_{1j} - R_{0j}, j = 0, \dots, n$$

- **řídicí body společné strany plátů leží ve středu úseček**, které spojují vždy předposlední bod řídicího polygonu plátu  $R$  ve směru  $u$  s druhým bodem plátu  $Q$  ve stejném směru
  - $G^1$  spojitost pokud je splněna lineární závislost s koeficientem  $k > 0$

$$Q_{sj} - Q_{s-1j} = k(R_{1j} - R_{0j}), j = 0, \dots, n$$

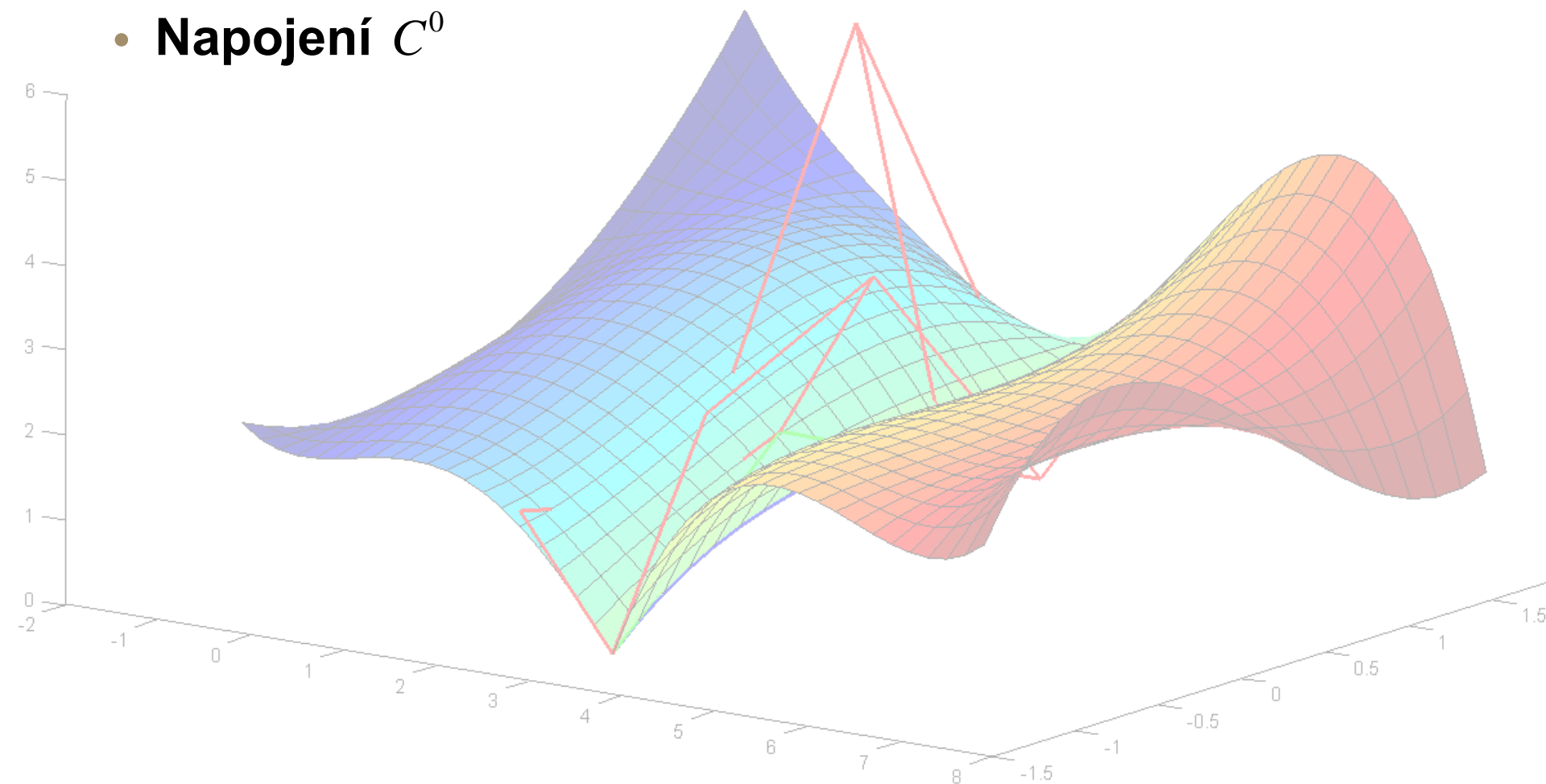
# Bézierovy plochy (12)

- Napojení  $C^1$



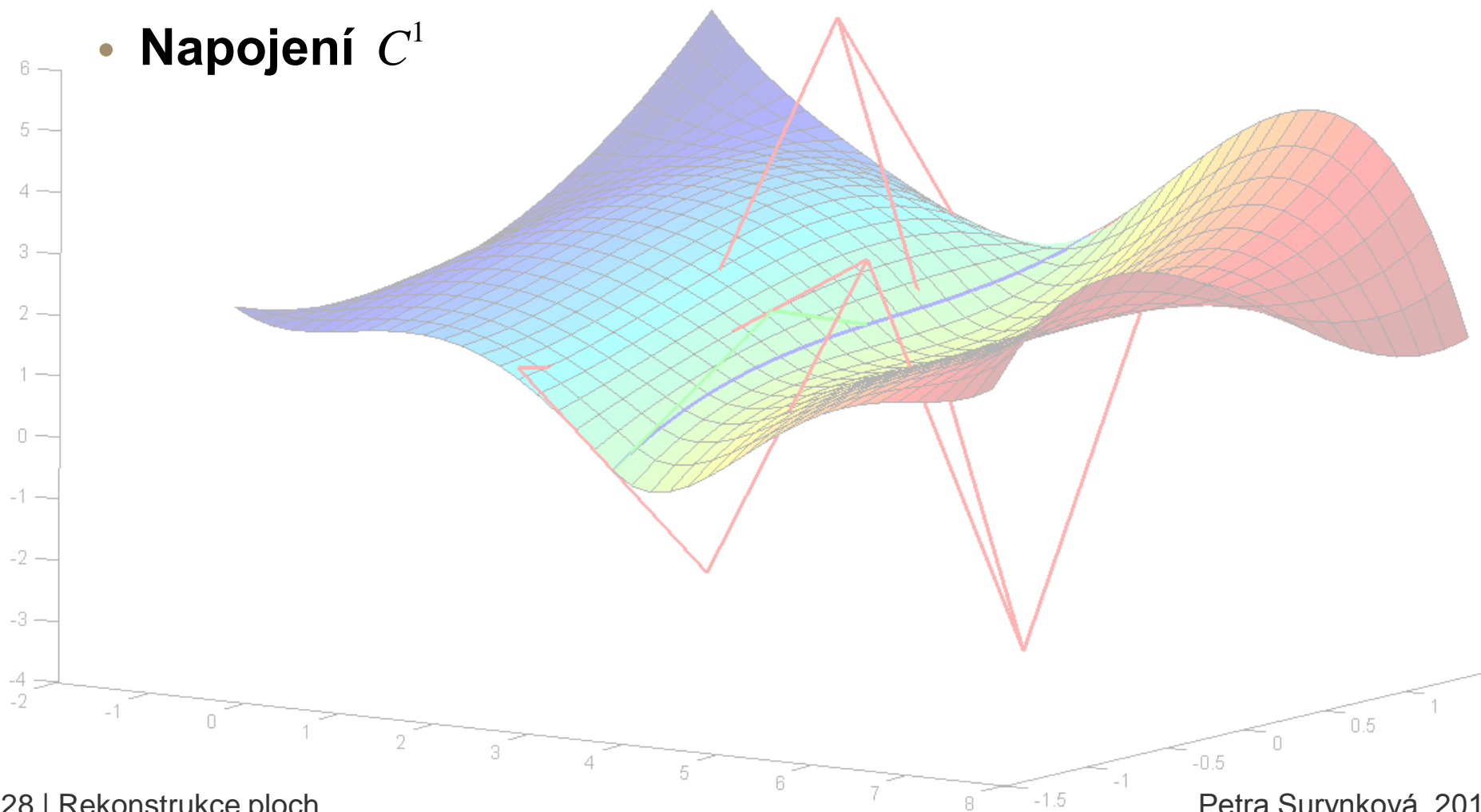
# Bézierovy plochy (13)

- Napojení  $C^0$



# Bézierovy plochy (14)

- Napojení  $C^1$



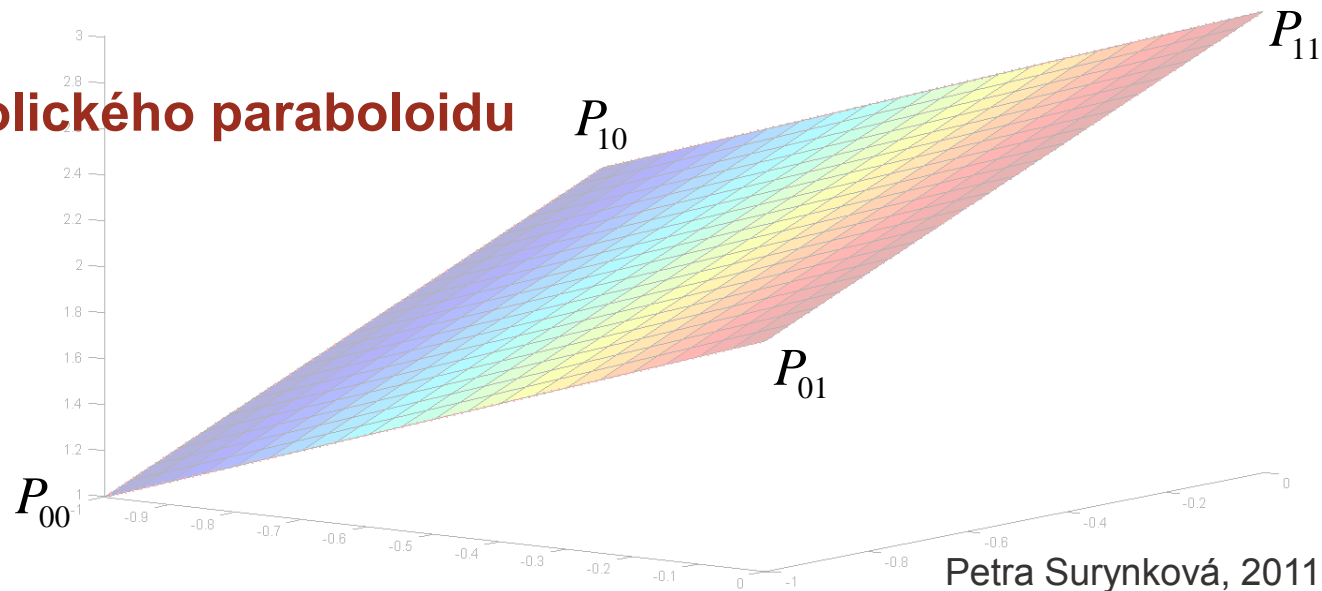


# Bézierovy plochy (15)

- **Speciální příklady Bézierových ploch**
  - **bilinéární Bézierova plocha** – tj.  $m = 1, n = 1$ 
    - je určena  $2 \times 2$  řídicími body
    - $u$ -křivky a  $v$ -křivky jsou pouze úsečky
  - výpočtem lze dokázat, že tímto způsobem lze definovat pouze **část roviny** (pokud řídicí body leží v jedné rovině)

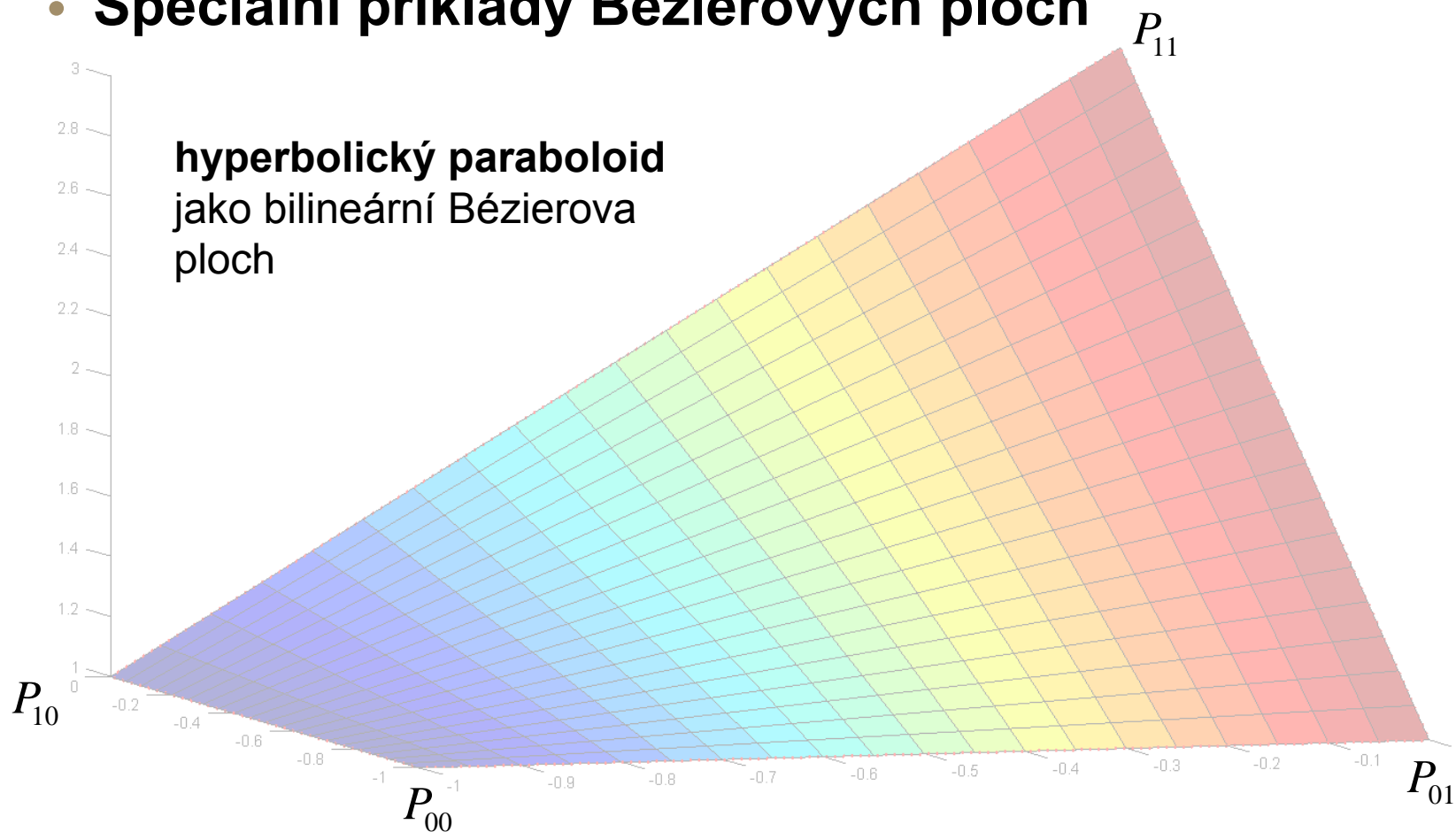
nebo

**část hyperbolického paraboloidu**



# Bézierovy plochy (16)

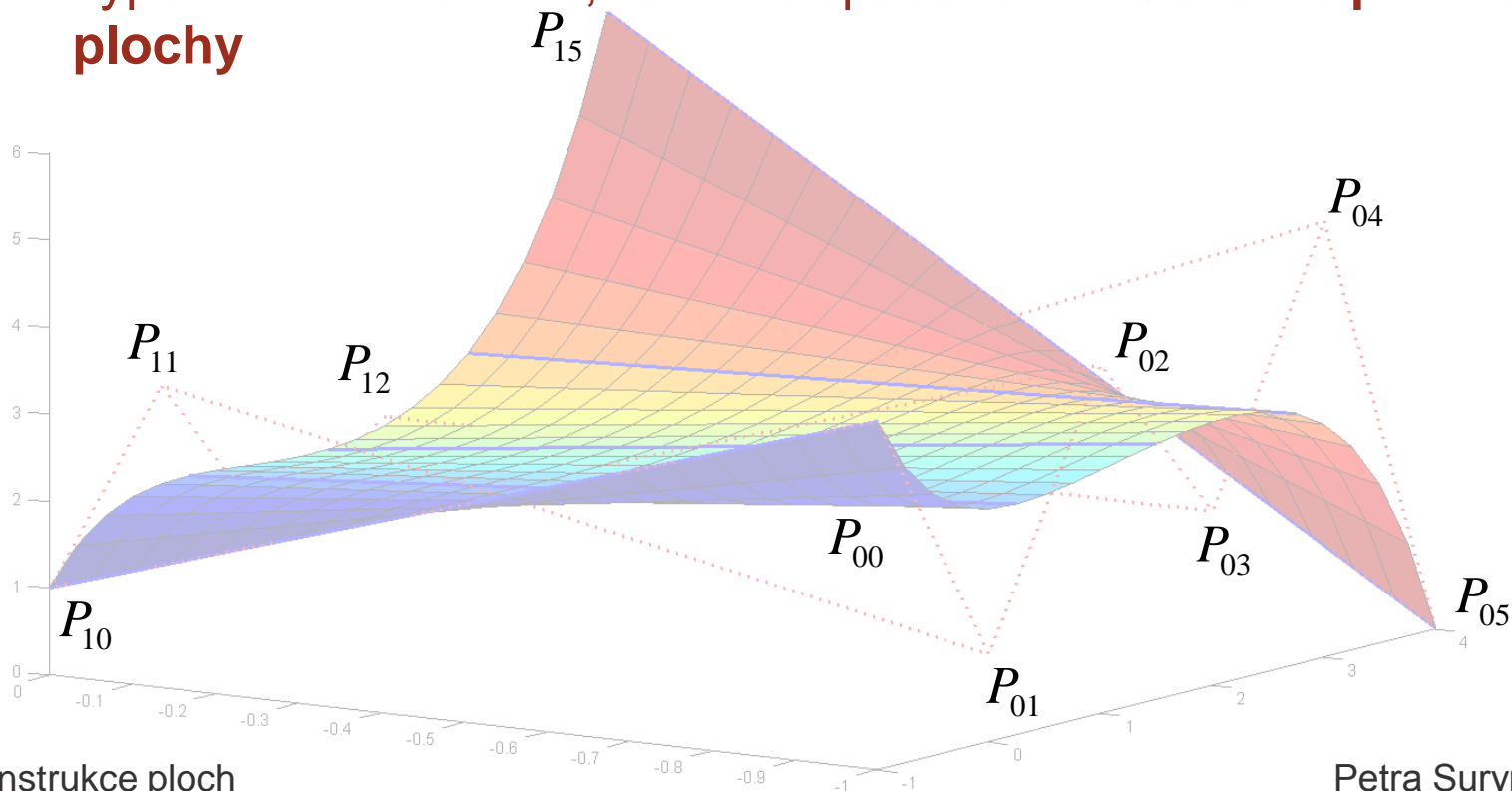
- Speciální příklady Bézierových ploch



# Bézierovy plochy (17)

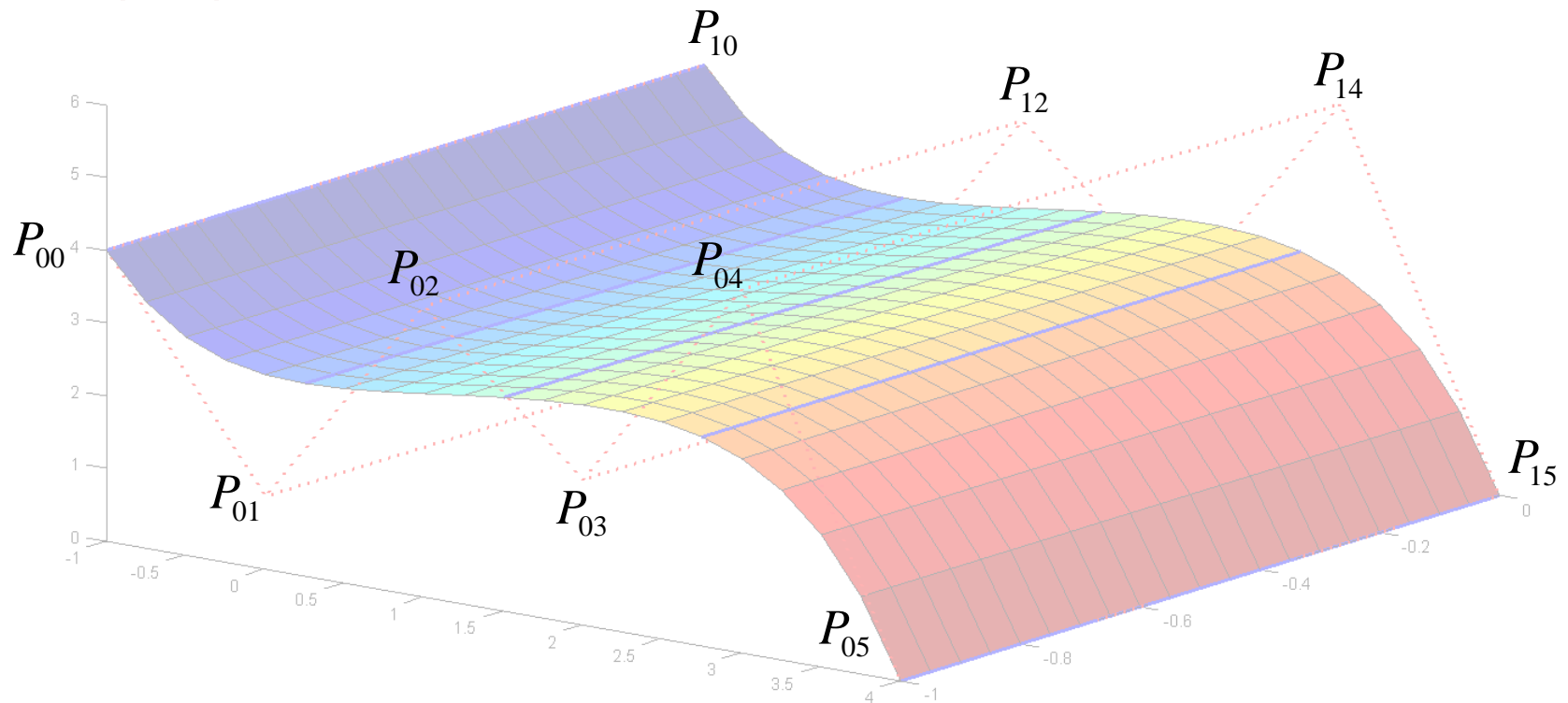
- Speciální příklady Bézierových ploch

- **Bézierova plocha  $m \times n$ , kde  $m = 1$**
- výpočtem lze dokázat, že tímto způsobem dostáváme **přímkové plochy**



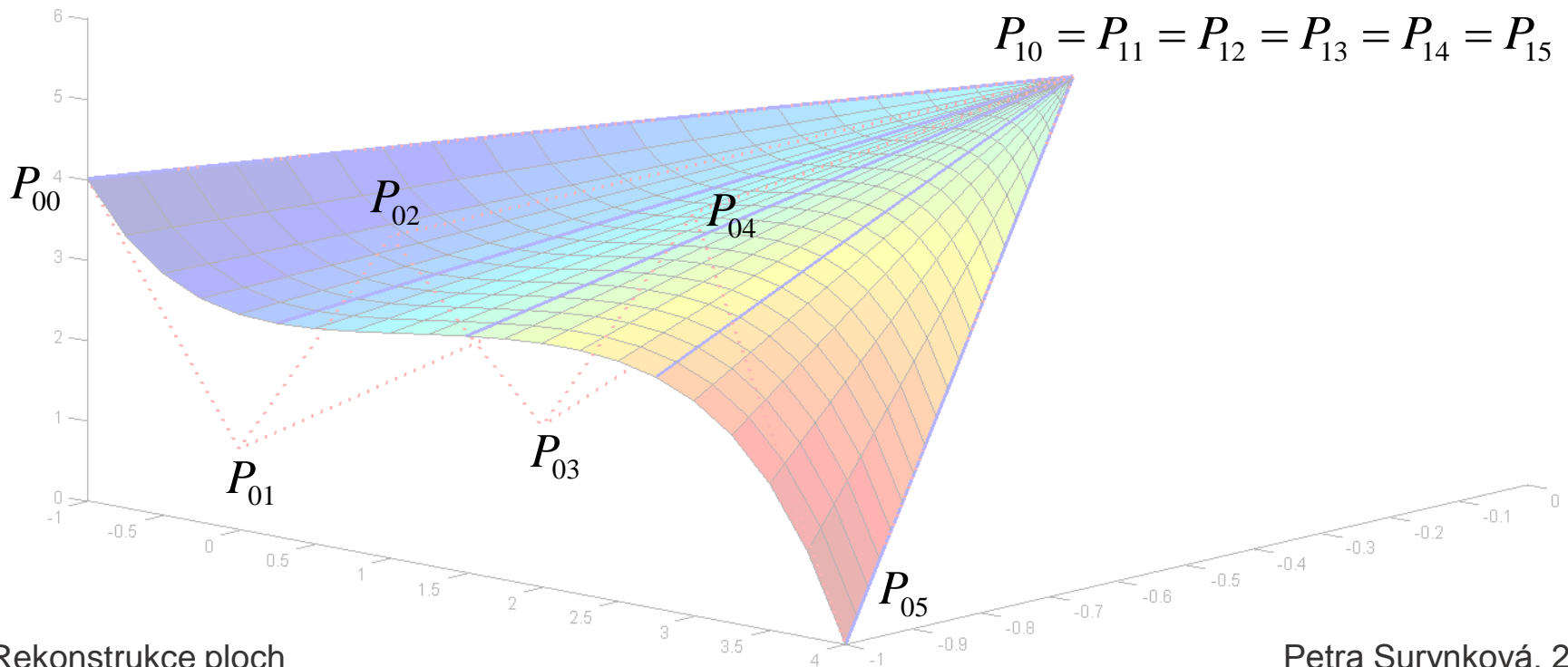
# Bézierovy plochy (18)

- Speciální příklady Bézierových ploch
  - Bézierova plocha  $m \times n$ , kde  $m = 1$
  - při speciální volbě řídicích bodů – **válcová plocha**



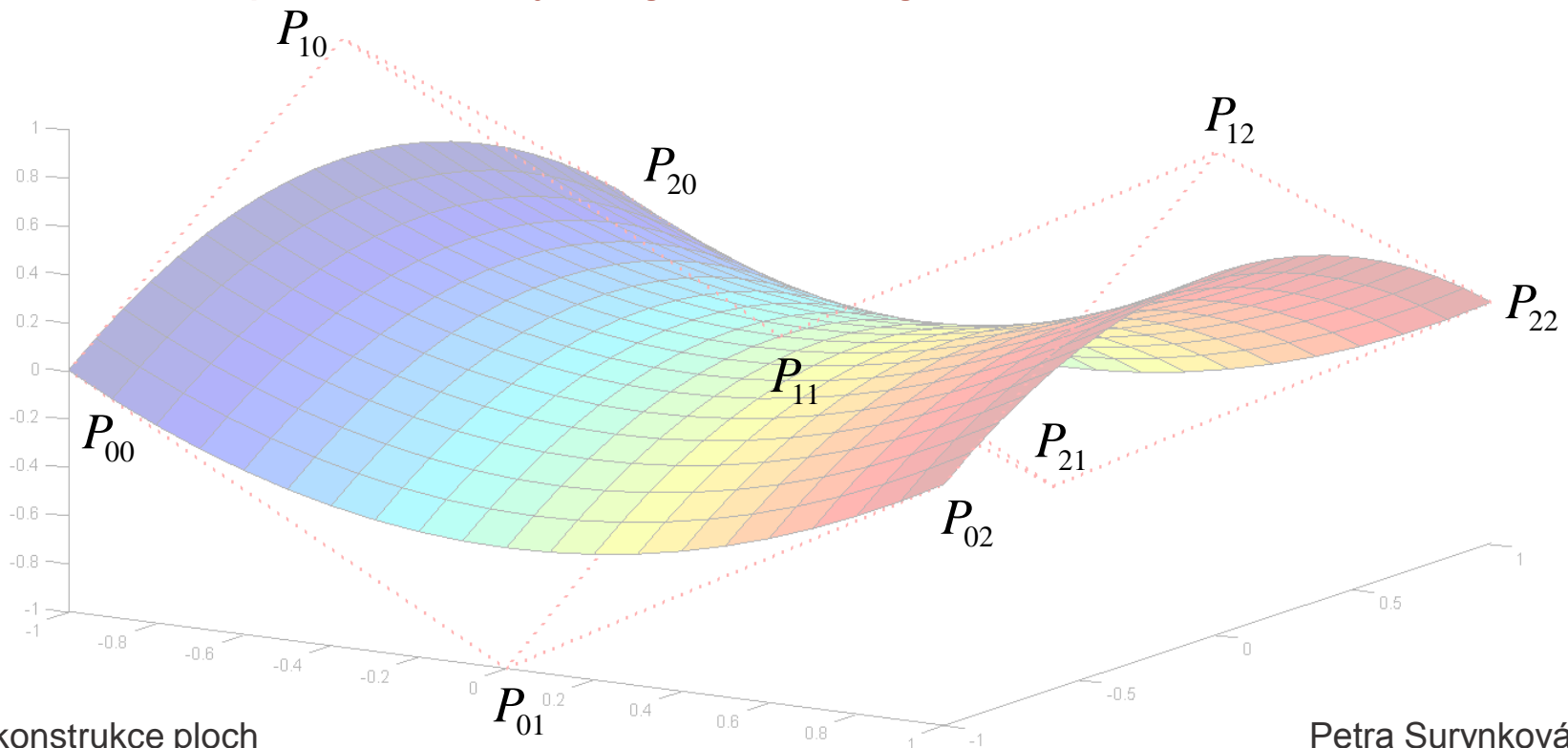
# Bézierovy plochy (19)

- Speciální příklady Bézierových ploch
  - Bézierova plocha  $m \times n$ , kde  $m = 1$
  - při speciální volbě řídicích bodů – kuželová plocha



# Bézierovy plochy (20)

- Speciální příklady Bézierových ploch
  - **Bézierova plocha  $m \times n$** , kde  $m = 2, n = 2$
  - další speciální volby – **hyperbolický paraboloid**



# Racionální Bézierovy plochy (1)

- K řídicím bodům jsou přiřazeny tzv. **váhy**  $w_{ij}$ 
  - jestliže jsou všechny váhy rovny jedné, přechází racionální Bézierova plocha v Bézierovu plochu
    - racionální Bézierovy plochy jsou zobecněním Bézierových ploch
    - **dovolují přesný popis kvadrik, lze popsat více speciálních ploch**
      - vychází z toho, že kromě paraboly nelze Bézierovými křivkami popsat kuželosečky
    - **pro modelování k dispozici další parametry, měníme tvar plochy bez změny řídicích bodů**

$$Q(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n w_{ij} P_{ij} B_i^m(u) B_j^n(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{ij} B_i^m(u) B_j^n(v)}$$

$$u \in \langle 0, 1 \rangle, v \in \langle 0, 1 \rangle$$

# Racionální Bézierovy plochy (2)

- Speciální příklady racionálních Bézierových ploch
  - př. ¼ rotační válcové plochy

$$P_{00} = [r, 0, 0] \quad w_{00} = 1$$

$$P_{10} = [r, 0, v] \quad w_{10} = 1$$

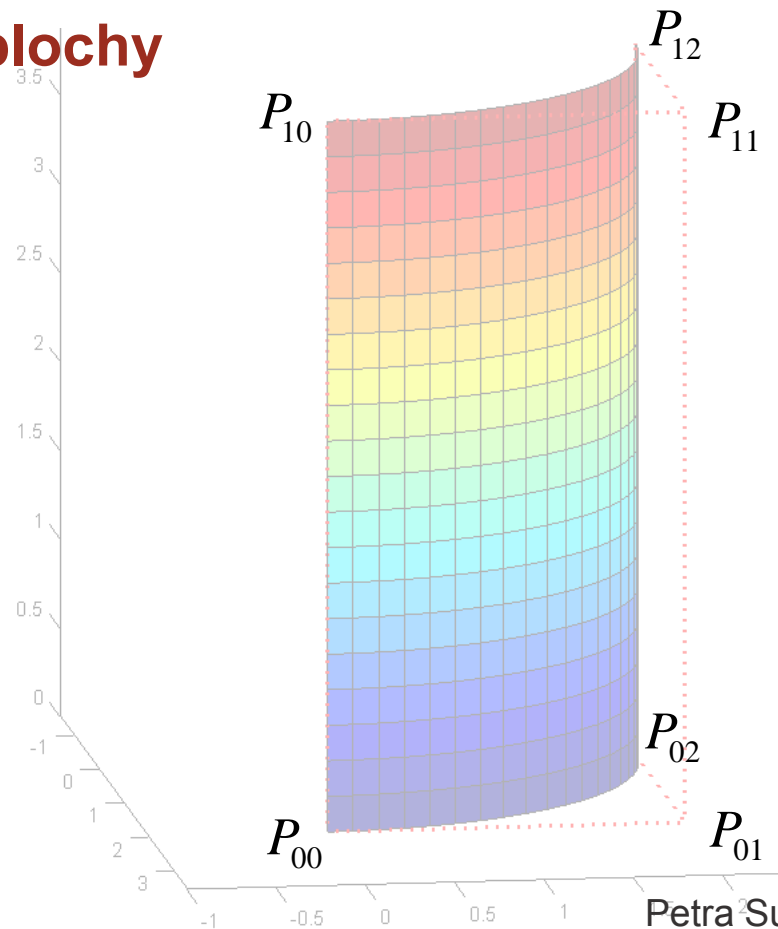
$$P_{01} = [r, r, 0] \quad w_{01} = \frac{\sqrt{2}}{2}$$

$$P_{11} = [r, r, v] \quad w_{11} = \frac{\sqrt{2}}{2}$$

$$P_{02} = [0, r, 0] \quad w_{02} = 1$$

$$P_{12} = [0, r, v] \quad w_{12} = 1$$

$v$  – výška,  $r$  – poloměr válce





# Racionální Bézierovy plochy (3)

- Speciální příklady racionálních Bézierových ploch

- př. 1/16 anuloidu

$$P_{00} = [0, R, r] \quad w_{00} = 1$$

$$P_{10} = [R, R, r] \quad w_{10} = \frac{\sqrt{2}}{2}$$

$$P_{20} = [R, 0, r] \quad w_{20} = 1$$

$$P_{01} = [0, R + r, r] \quad w_{01} = \frac{\sqrt{2}}{2}$$

$$P_{11} = [R + r, R + r, r] \quad w_{11} = \frac{1}{2}$$

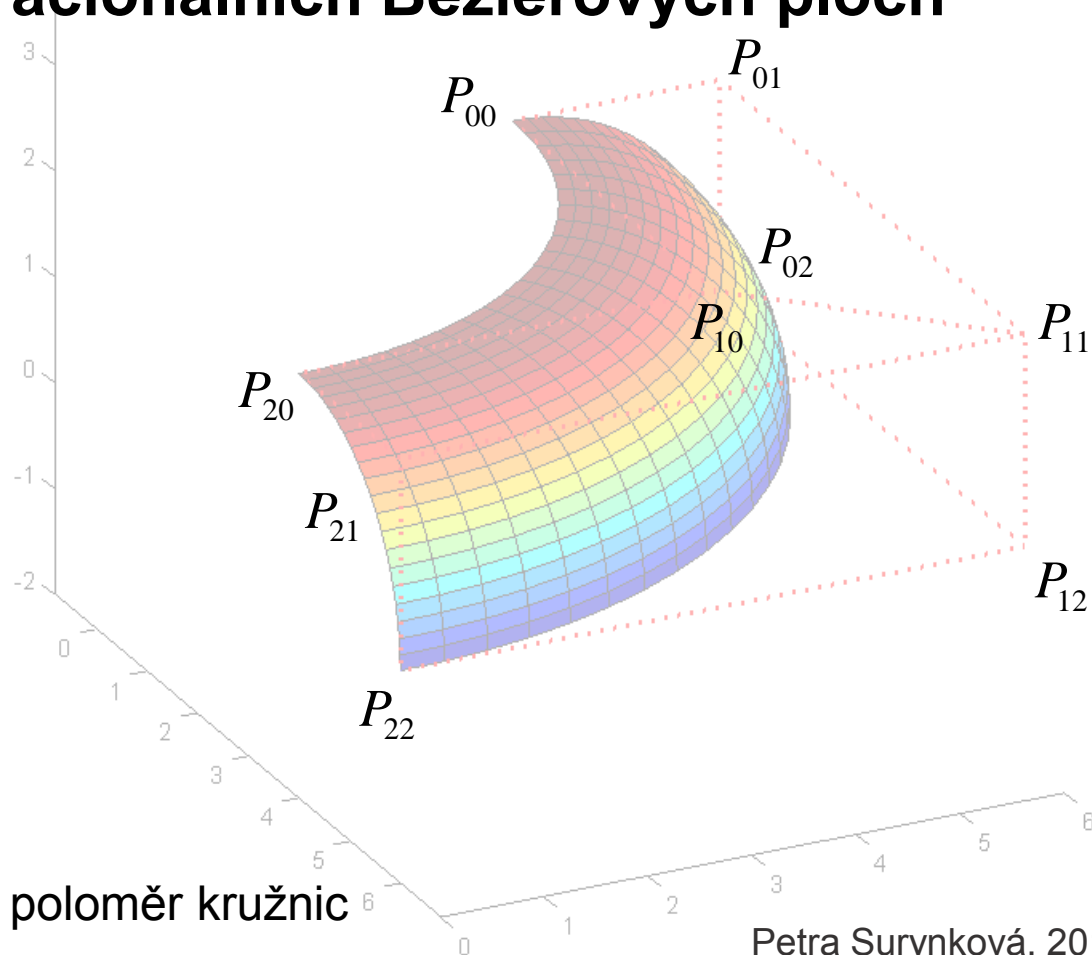
$$P_{21} = [R + r, 0, r] \quad w_{21} = \frac{\sqrt{2}}{2}$$

$$P_{02} = [0, R + r, 0] \quad w_{02} = 1$$

$$P_{12} = [R + r, R + r, 0] \quad w_{12} = \frac{\sqrt{2}}{2}$$

$$P_{22} = [R + r, 0, 0] \quad w_{22} = 1$$

$r, R$  – poloměr kružnic



# Racionální Bézierovy plochy (4)

- Speciální příklady racionálních Bézierových ploch

- př. 1/8 koule

$$P_{00} = [0, 0, r] \quad w_{00} = 1$$

$$P_{10} = [r, 0, r] \quad w_{10} = \frac{\sqrt{2}}{2}$$

$$P_{20} = [r, 0, 0] \quad w_{20} = 1$$

$$P_{01} = [0, 0, r] \quad w_{01} = 1$$

$$P_{11} = [r, r, r] \quad w_{11} = \frac{1}{2}$$

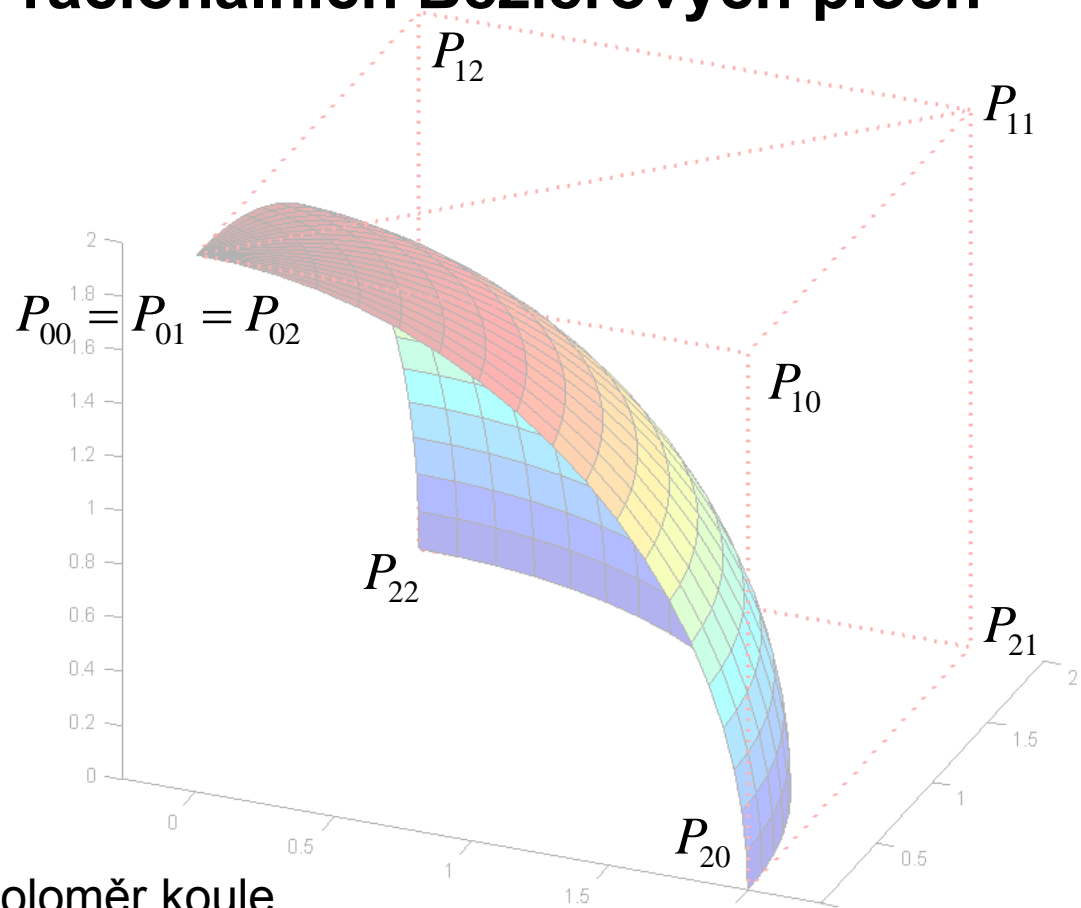
$$P_{21} = [r, r, 0] \quad w_{21} = \frac{\sqrt{2}}{2}$$

$$P_{02} = [0, 0, r] \quad w_{02} = 1$$

$$P_{12} = [0, r, r] \quad w_{12} = \frac{\sqrt{2}}{2}$$

$$P_{22} = [0, r, 0] \quad w_{22} = 1$$

$r$  - poloměr koule



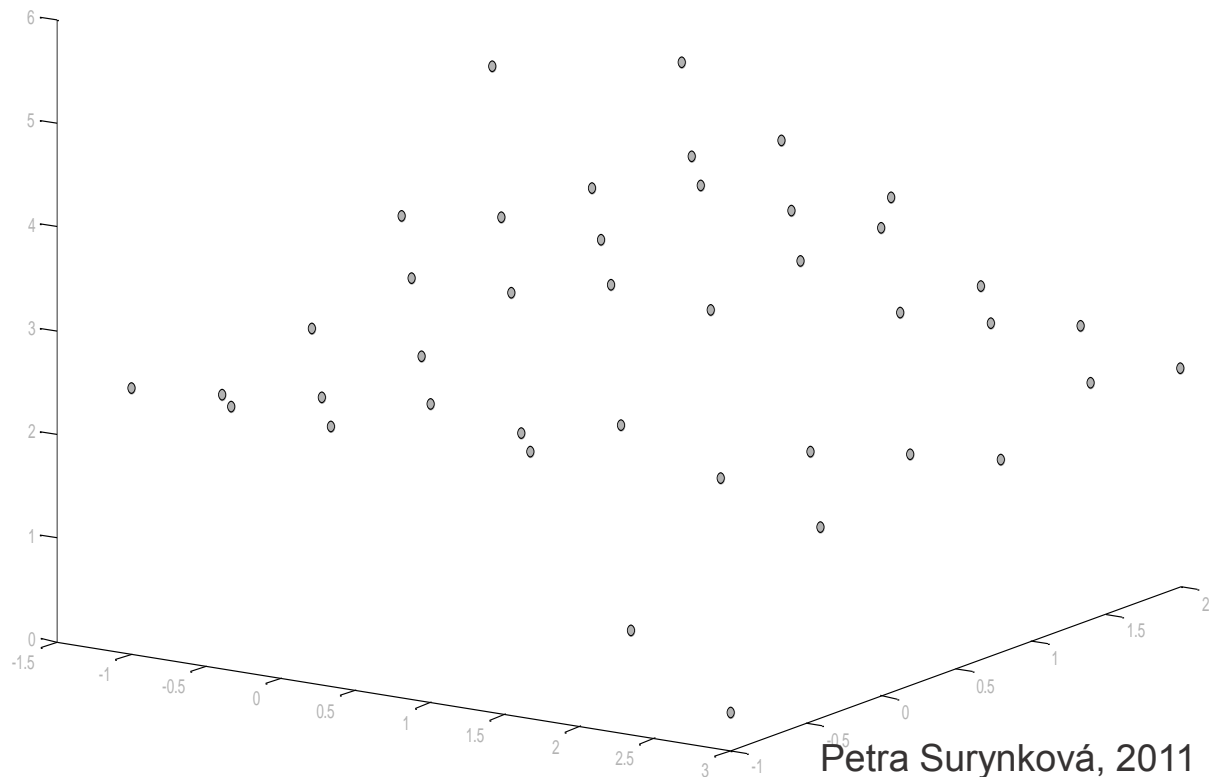
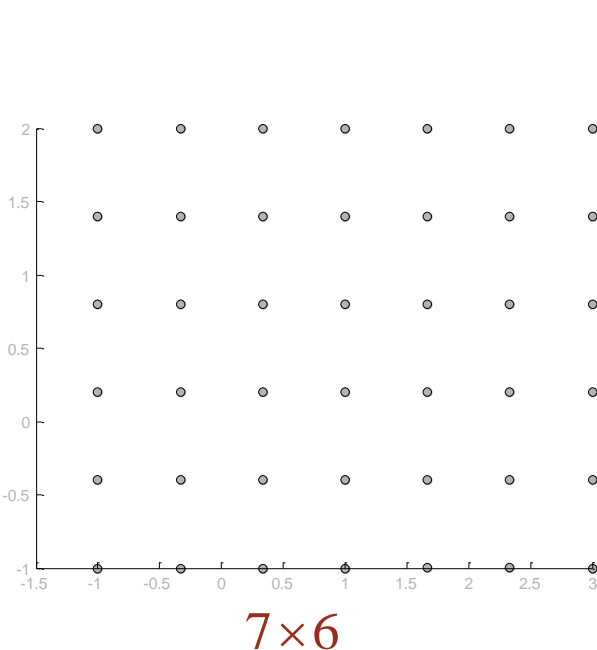
# Programová demonstrace



**Ukázka obecných racionálních Bézierových  
plátů v prostředí Matlab**

# Aproximace ploch Bézierovými pláty (1)

- Vstupní množina – **body v prostoru**, známe pouze souřadnice
- Zjednodušení – **body tvoří v půdoryse pravidelnou obdélníkovou mřížku**



# Aproximace ploch Bézierovými pláty (2)

- Vstupní množina – **body v prostoru**, známe pouze souřadnice
- Zjednodušení – **body tvoří v půdoryse pravidelnou obdélníkovou mřížku**

$$X(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_i^m(u) B_j^n(v), u \in \langle 0, 1 \rangle, v \in \langle 0, 1 \rangle$$

známe

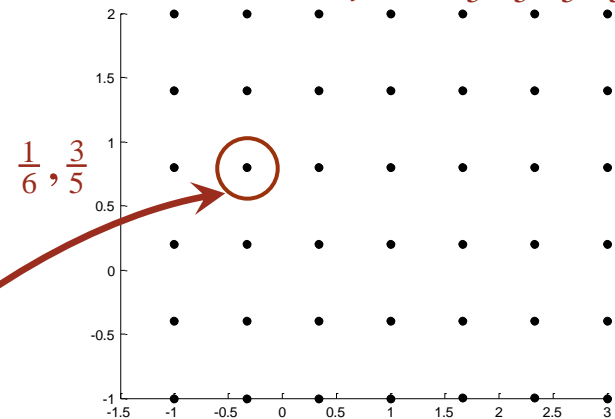
hledáme

$$u_k = 0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1$$

$$v_l = 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1$$

- bodům přiřadíme parametry podle mřížky

$$\square \text{ tj. } X(u_k, v_l) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_i^m(u_k) B_j^n(v_l)$$



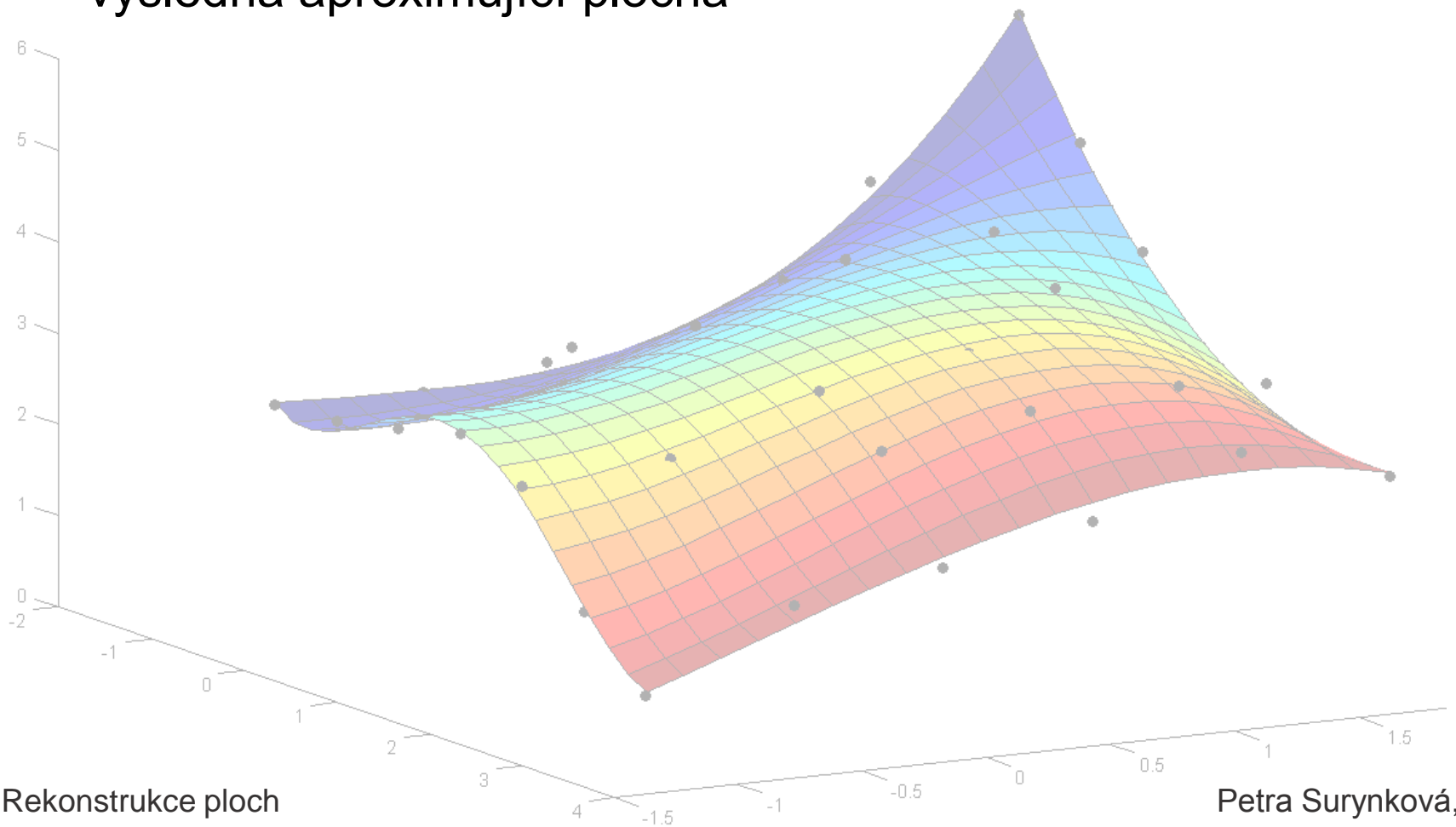
# Aproximace ploch Bézierovými pláty (3)

- **stupeň plochy**  $m \times n$  volíme
  - podle toho, zda chceme plochu interpolovat nebo aproximovat
  - pokud máme o ploše další informace – např. jde o body klenby, můžeme předpokládat nízký stupeň plochy, nebo stupeň dokonce známe
- hledáme **řídící body**  $P_{ij}$  **ve smyslu metody nejmenších čtverců**
  - využíváme funkcí Matlabu k řešení přeuredené soustavy rovnic
- **jedna rovnice soustavy**

$$X(u_k, v_l) = \left( B_0^m(u_k), B_1^m(u_k), \dots, B_m^m(u_k) \right) P \left( B_0^n(v_l), B_1^n(v_l), \dots, B_n^n(v_l) \right)^T$$

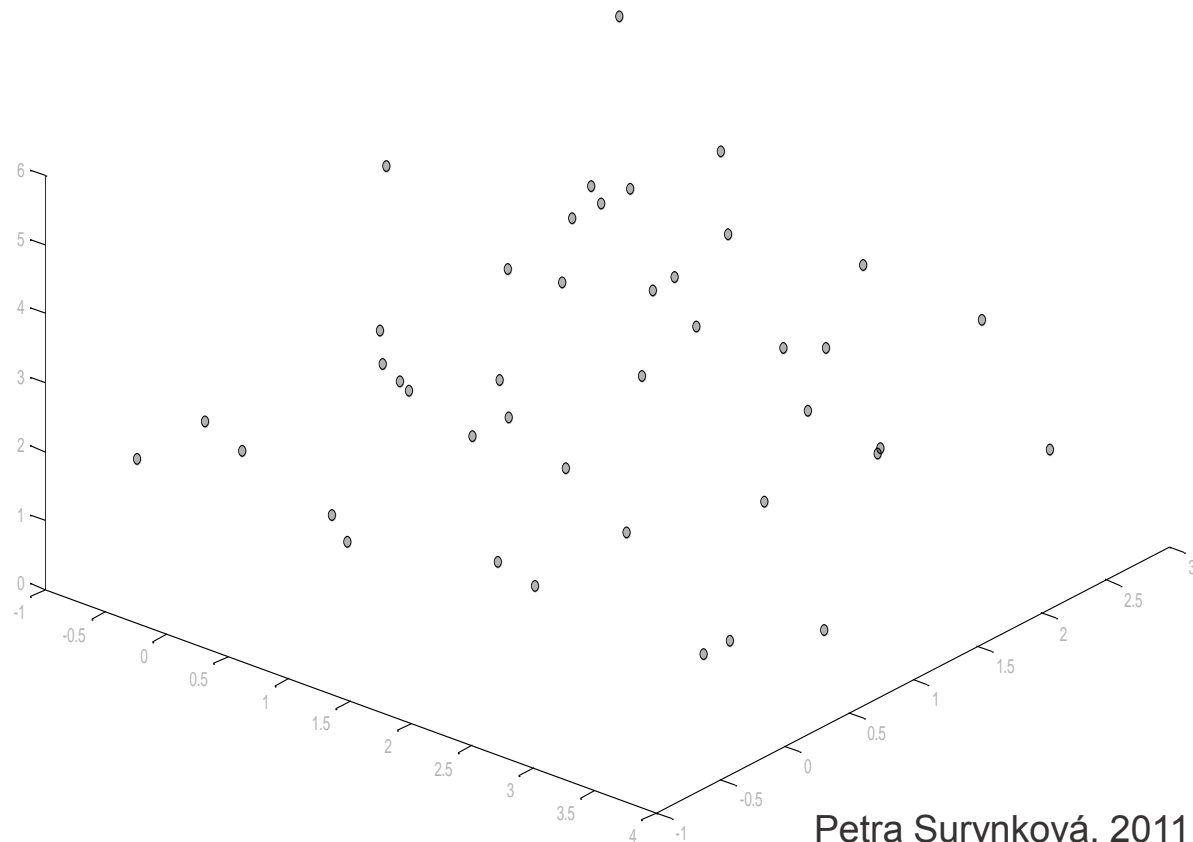
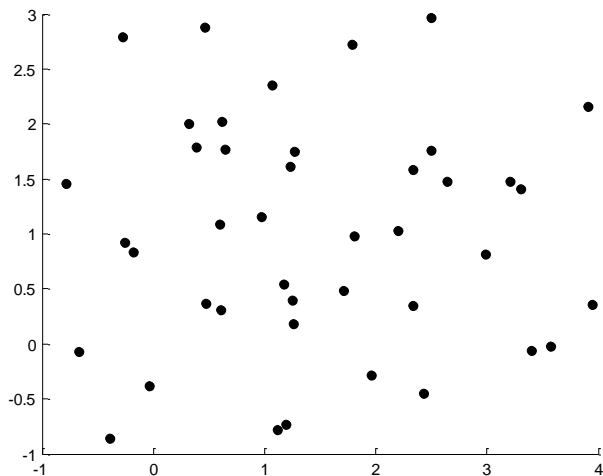
# Aproximace ploch Bézierovými pláty (4)

- výsledná aproximující plocha



# Aproximace ploch Bézierovými pláty (5)

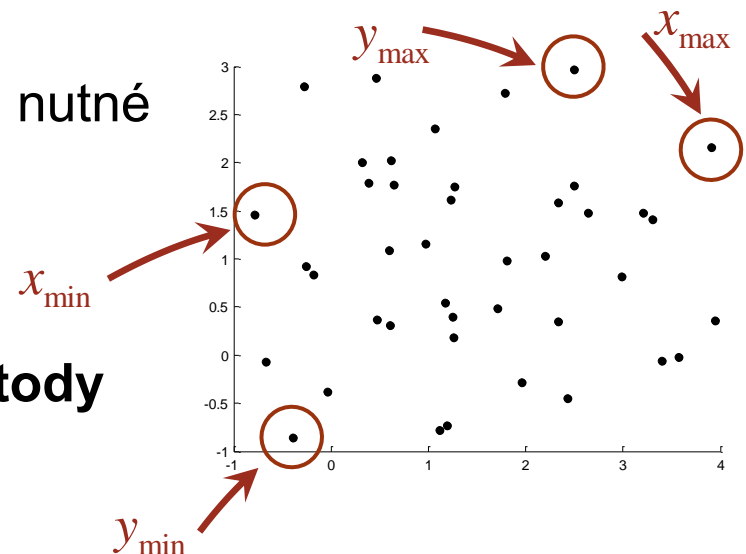
- Vstupní množina – **body v prostoru**, známe pouze souřadnice
- Obecné – **body netvoří v půdoryse pravidelnou mřížku**





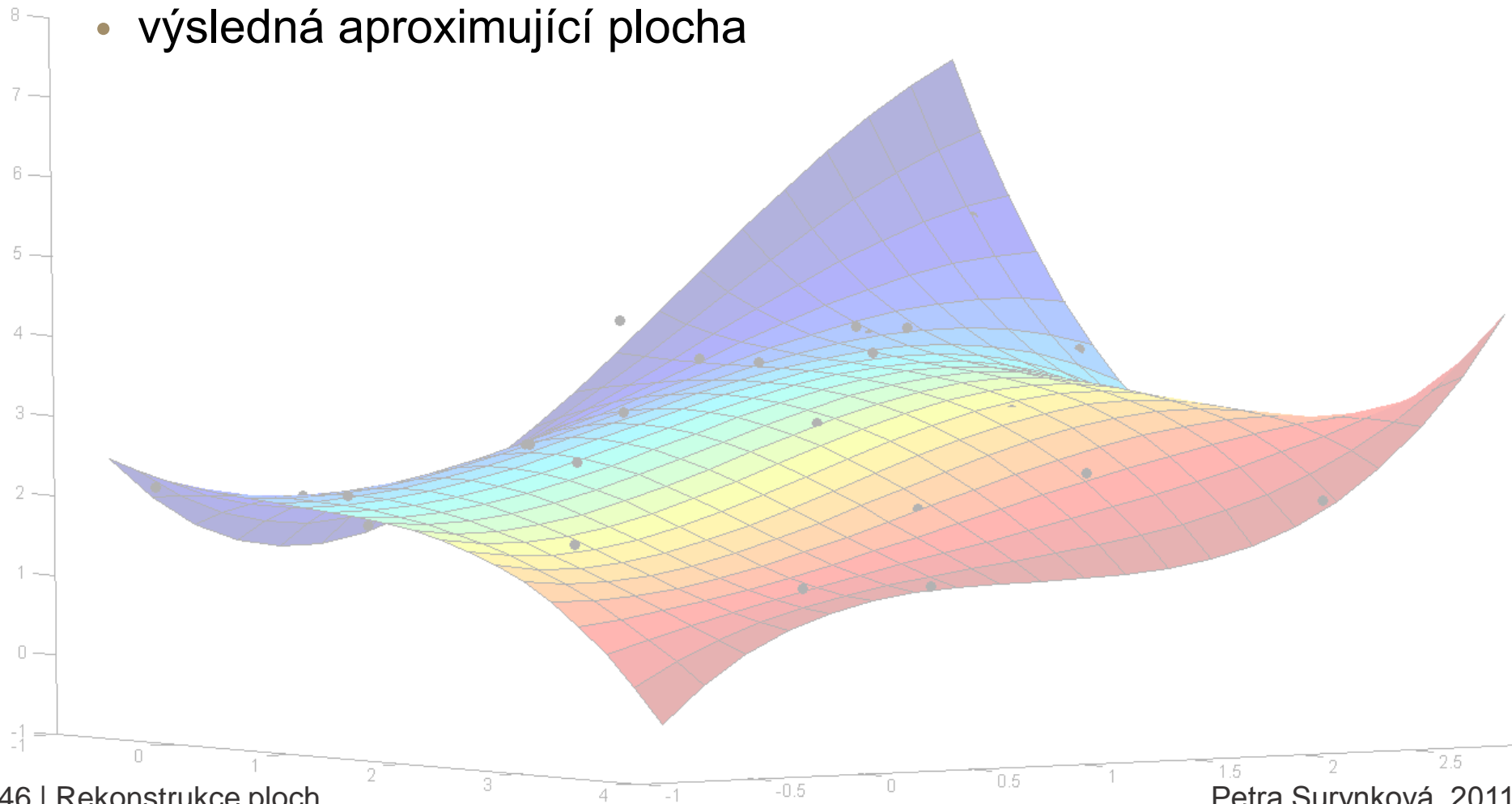
# Aproximace ploch Bézierovými pláty (6)

- Vstupní množina – **body v prostoru**, známe pouze souřadnice
- Obecné – **body netvoří v půdoryse pravidelnou mřížku**
- bodům přiřadíme parametry podle přeškálování
  - určujeme body s minimální a maximální x-ovou a y-ovou souřadnicí
- kvůli nepravidelnosti vstupní množiny je nutné body přeuspořádat do matic
- opět **stupeň plochy**  $m \times n$  volíme
- hledáme **řídící body**  $P_{ij}$  ve smyslu metody **nejmenších čtverců**



# Aproximace ploch Bézierovými pláty (7)

- výsledná aproximující plocha



# Programová demonstrace



**Ukázka aproximace Bézierovými pláty v  
prostředí Matlab**

# Další vylepšování tvaru (1)

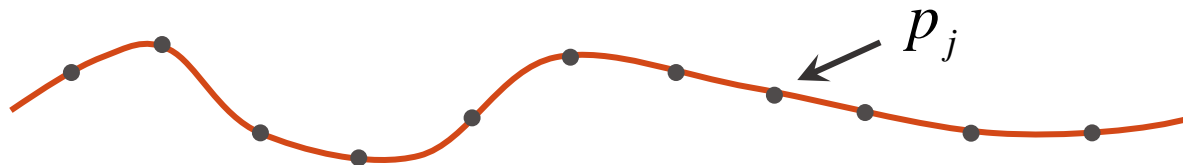
- Snaha přinést do zpracování rekonstrukce povrchů nové metody
- Pojem postupného vylepšování tvaru vysvětlíme na křivkách
  - ukázka **evoluce uzavřených a otevřených B-spline křivek**
- Cílem je vhodně aproximovat zadanou množinu bodů v rovině daným typem křivky
  - zvolená metoda je evoluce
- **Podstata evoluce**
  - postupný vývoj křivky od jistého počátečního tvaru a polohy
  - křivka je tomuto procesu podrobena do té doby, dokud nesplňuje nějaké předem dané kritérium
  - zde, dokud **neprochází body vstupní množiny** (nebo vhodně aproximuje vstupní množinu bodů)

# Další vylepšování tvaru (2)

- dána množina bodů  $\{p_j\}_{j=1..N}$  v rovině
- cílem je proložit tyto body křivkou
- výslednou křivku hledáme tak, aby byla splněna podmínka

$$\sum_{j=1}^N \min_{x_j \in c} \|p_j - x_j\|^2 \rightarrow \min,$$

kde  $x_j$  značí body na křivce



# Další vylepšování tvaru (3)

- mějme **parametricky popsanou křivku**

$$c(u) = \sum_{i=0}^m B_i(u) \cdot V_i$$

$$u \in [a, b]$$


$u$  **parametr křivky**

$V_i$  **řídící body** křivky

$B_i$  **bázové funkce**

- označme  $c_s(u) := c(s, u)$ 
  - v reprezentaci křivky se objevují dva rozdílné parametry

**vektor parametrů**  $s = (s_1, s_2, \dots, s_n)$   
 $s \in \Omega \subset \mathbb{R}^n$

  $\sum_{j=1}^N \min_{u_j} \|p_j - c_s(u_j)\|^2 \rightarrow \min$

- $s_j$  jsou x-ové a y-ové souřadnice řídících bodů

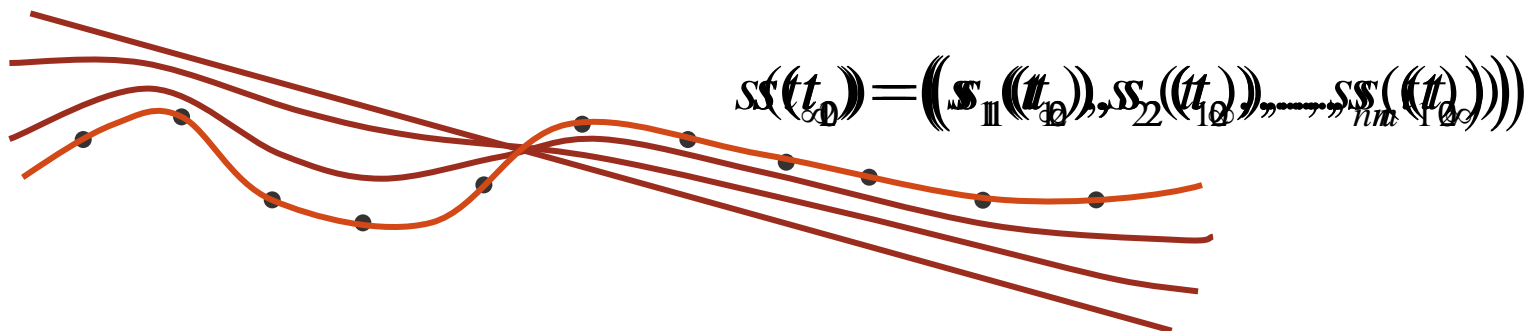
# Další vylepšování tvaru (4)

- hledáme vektor parametrů  $s = (s_1, s_2, \dots, s_n)$ , který definuje křivku
- nechť vektor parametrů závisí na dalším parametru  $t$  – **evoluční parametr**

- parametr  $t$  můžeme chápat jako čas

⇒  $s(t) = (s_1(t), s_2(t), \dots, s_n(t))$

- tyto **parametry** jsou v čase **modifikovány** tak, že se počáteční **křivka mění a pohybuje stále blíž k daným bodům**



# Další vylepšování tvaru (5)

- jak je řízen pohyb

- pro každý bod  $\{p_j\}_{j=1..N}$  vstupní množiny spočítáme nejbližší bod na křivce  $f_j = c(u_j)$
- během evoluce se body  $f_j$  pohybují k bodům vstupní množiny

## rychlostí

$$v_s(u_j) = \dot{c}_s(u_j) = \sum_{i=1}^n \frac{\partial c_s(u_j)}{\partial s_i} \dot{s}_i$$

nebo

## normálovou rychlostí

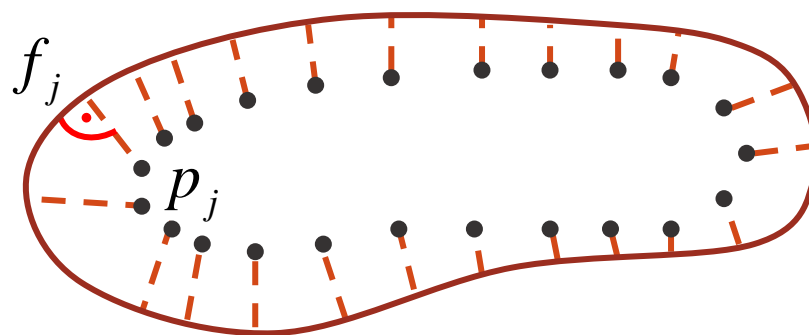
$$\left(v_s(u_j)\right)^{\top} n_s(u_j) = \sum_{i=1}^n \left(\frac{\partial c_s(u_j)}{\partial s_i} \dot{s}_i\right)^{\top} n_s(u_j)$$



# Další vylepšování tvaru (6)

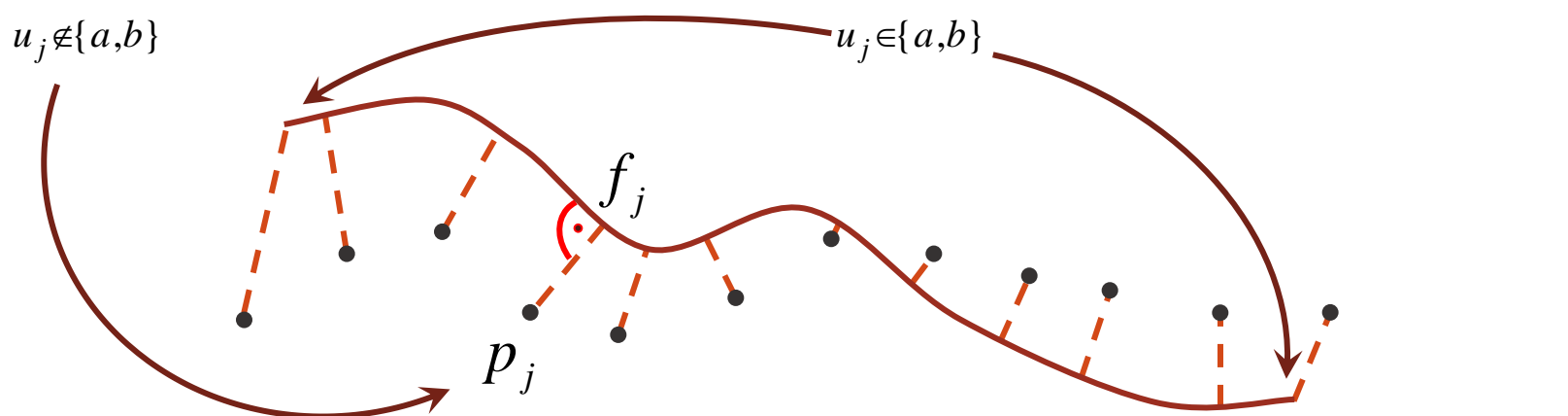
- jak je řízen pohyb
  - $n_s(u_j)$  značí jednotkovou normálu křivky v bodě  $c_s(u_j)$
  - označíme:  $d_j := p_j - f_j$
  - chceme **minimalizovat**

$$\sum_{j=1}^N \left\| \left( v_s(u_j) - d_j \right)^\top n_s(u_j) \right\|^2 \rightarrow \min_{\dot{s}}$$



# Další vylepšování tvaru (7)

- jak je řízen pohyb
  - $n_s(u_j)$  značí jednotkovou normálu křivky v bodě  $c_s(u_j)$
  - označíme:  $d_j := p_j - f_j$
  - chceme **minimalizovat**

$$\sum_{j=1}^N \left\| \left( v_s(u_j) - d_j \right)^\top n_s(u_j) \right\|^2 + \sum_{j=1}^N \left( v_s(u_j) - d_j \right)^2 \rightarrow \min_{\dot{s}}$$


The diagram illustrates the minimization of a functional for curve deformation. A solid brown curve is shown with several points marked by black dots. Dashed orange lines connect these points to a set of points labeled  $p_j$ . A red dot labeled  $f_j$  is also shown. Arrows indicate the direction of movement for the points. Labels  $u_j \notin \{a,b\}$  and  $u_j \in \{a,b\}$  are placed near the curve's ends.


# Další vylepšování tvaru (8)

- jak je řízen pohyb

- $n_s(u_j)$  značí jednotkovou normálu křivky v bodě  $c_s(u_j)$
- označíme:  $d_j := p_j - f_j$
- chceme **minimalizovat**

$$\sum_{\substack{j=1 \\ u_j \notin \{a,b\}}}^N \left\| \left( v_s(u_j) - d_j \right)^\top n_s(u_j) \right\|^2 + \sum_{u_j \in \{a,b\}} \left( v_s(u_j) - d_j \right)^2 \rightarrow \min_{\dot{s}}$$

- **výsledkem**  $(\dot{s}_1, \dot{s}_2, \dots, \dot{s}_n)$
- **aktualizujeme vektor parametrů**

  $(s_1 + \varepsilon \dot{s}_1, s_2 + \varepsilon \dot{s}_2, \dots, s_n + \varepsilon \dot{s}_n)$

# Programová demonstrace



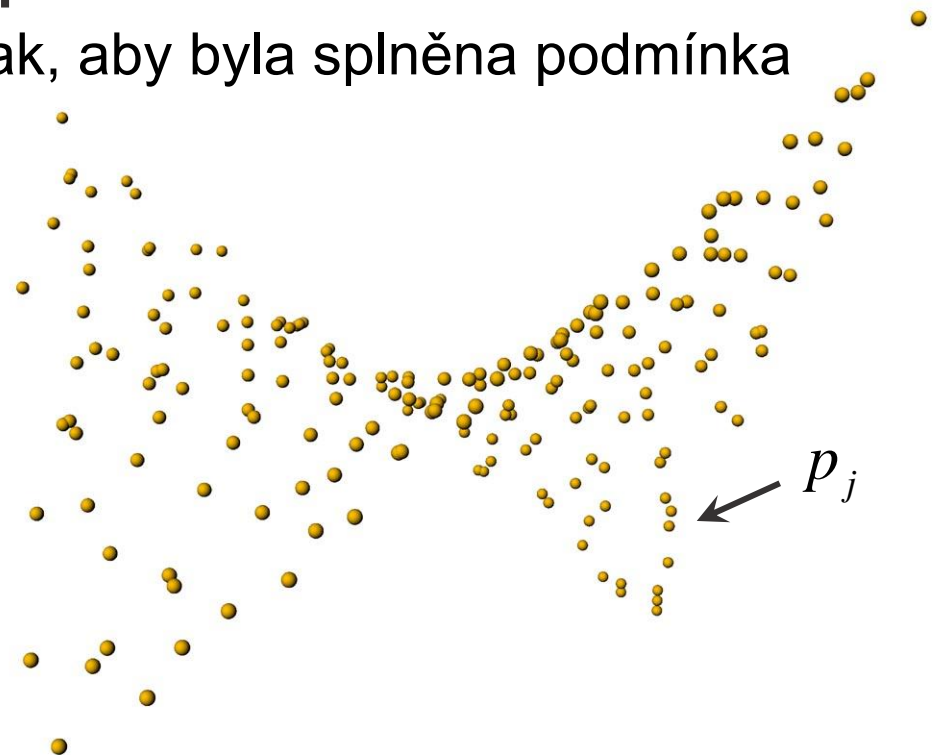
**Ukázka evoluce uzavřených a otevřených  
B-spline křivek v prostředí Matlab**

# Další vylepšování tvaru (9)

- dána množina bodů  $\{p_j\}_{j=1..N}$  v prostoru
- cílem je proložit tyto body plochou
- výslednou plochu hledáme tak, aby byla splněna podmínka

$$\sum_{j=1}^N \min_{x_j \in c} \|p_j - x_j\|^2 \rightarrow \min,$$

kde  $x_j$  značí body na ploše

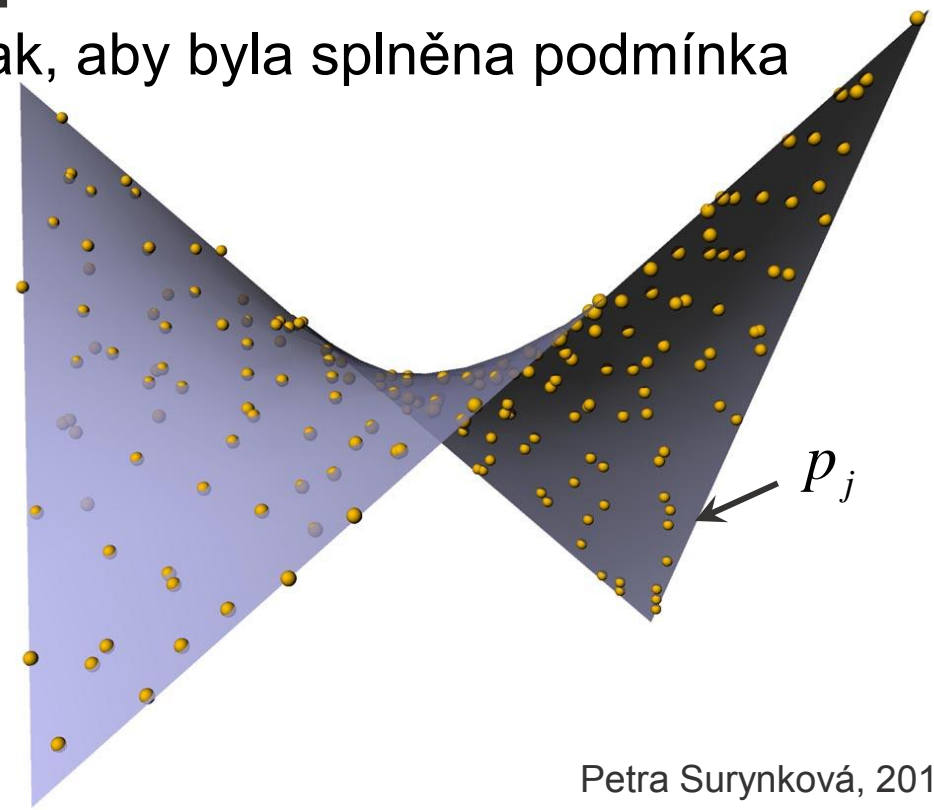


# Další vylepšování tvaru (10)

- dána množina bodů  $\{p_j\}_{j=1..N}$  v prostoru
- cílem je proložit tyto body plochou
- výslednou plochu hledáme tak, aby byla splněna podmínka

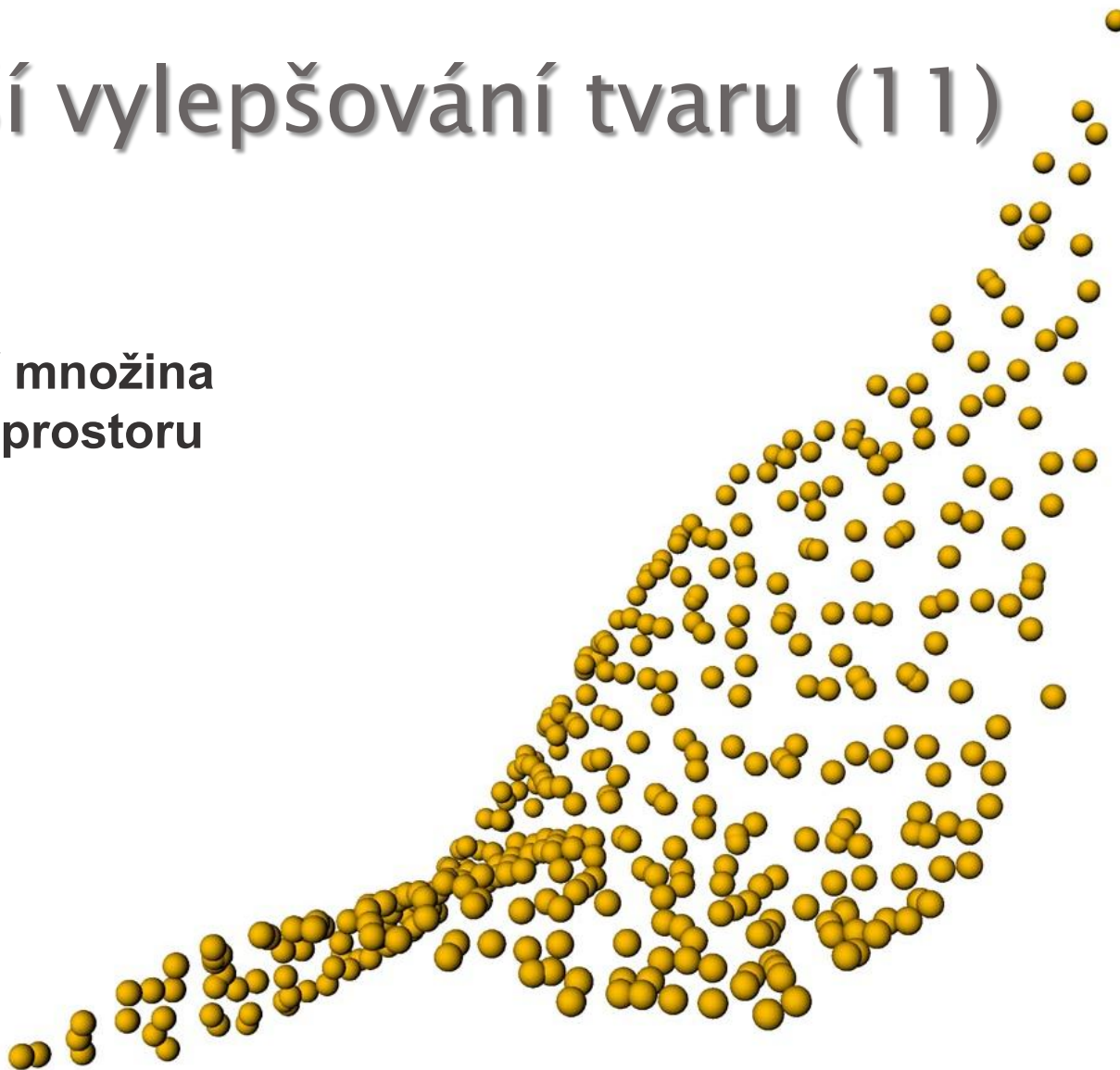
$$\sum_{j=1}^N \min_{x_j \in c} \|p_j - x_j\|^2 \rightarrow \min,$$

kde  $x_j$  značí body na ploše



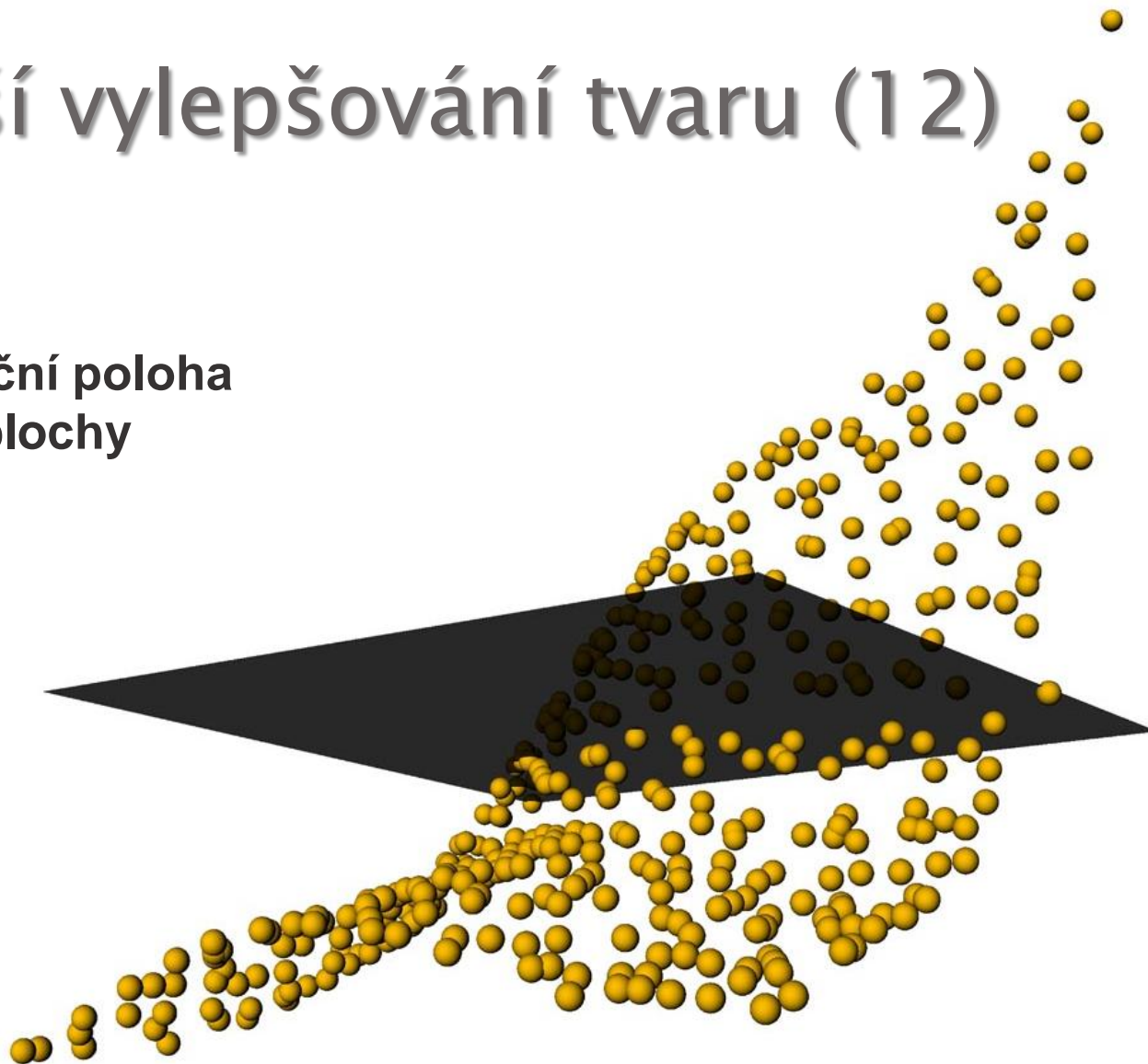
# Další vylepšování tvaru (11)

vstupní množina  
bodů v prostoru



# Další vylepšování tvaru (12)

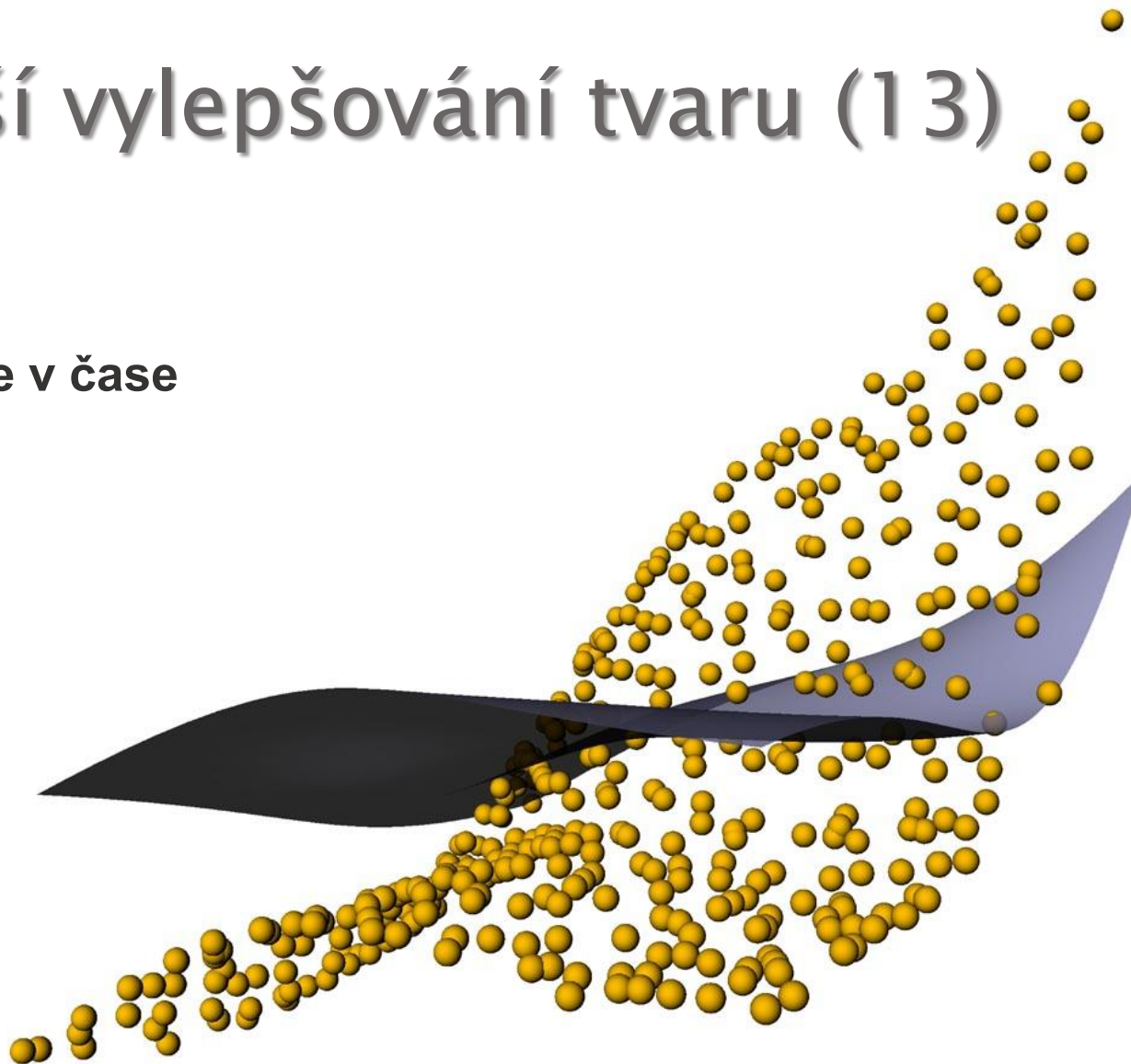
počáteční poloha  
a tvar plochy





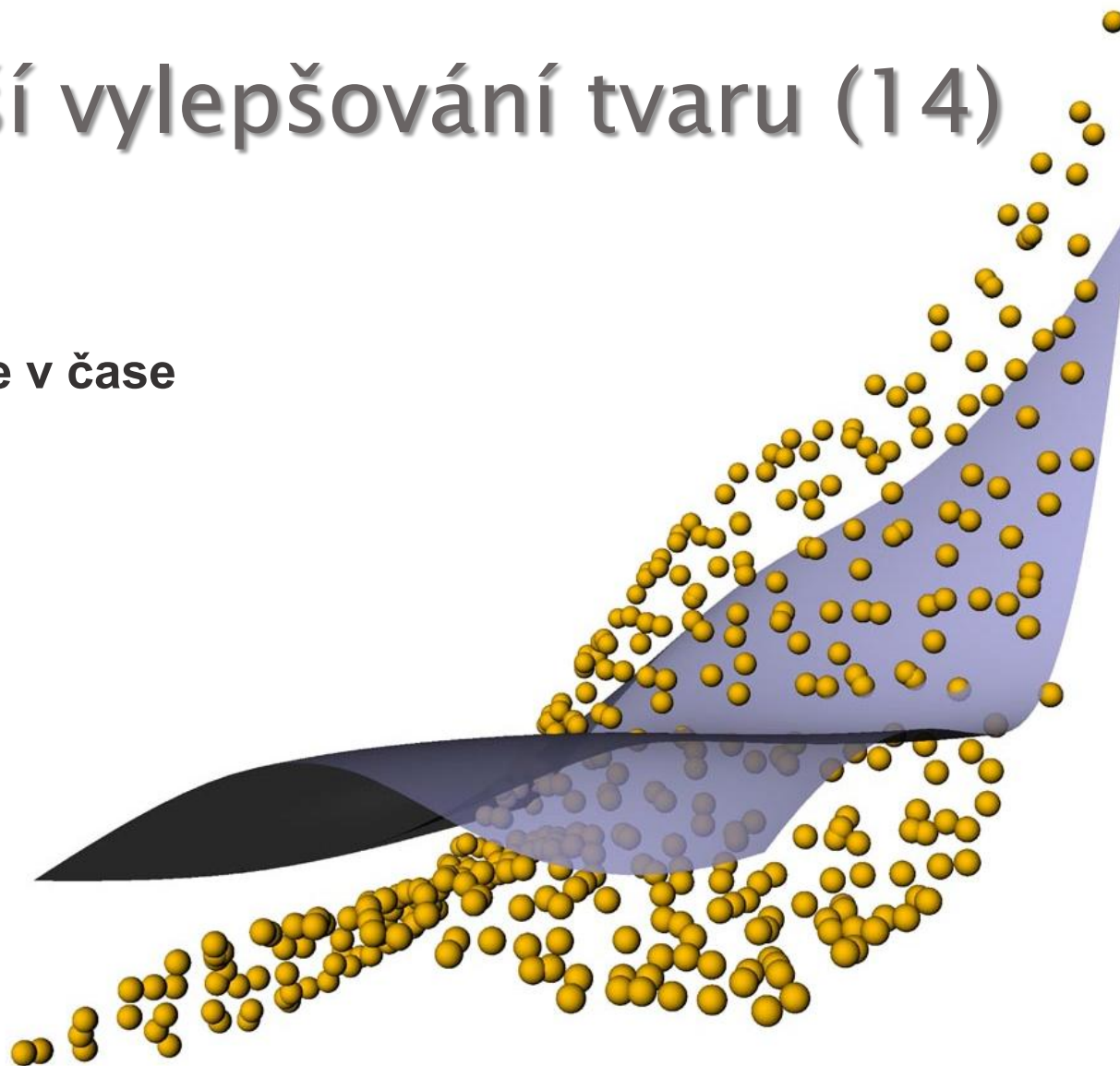
# Další vylepšování tvaru (13)

evoluce v čase



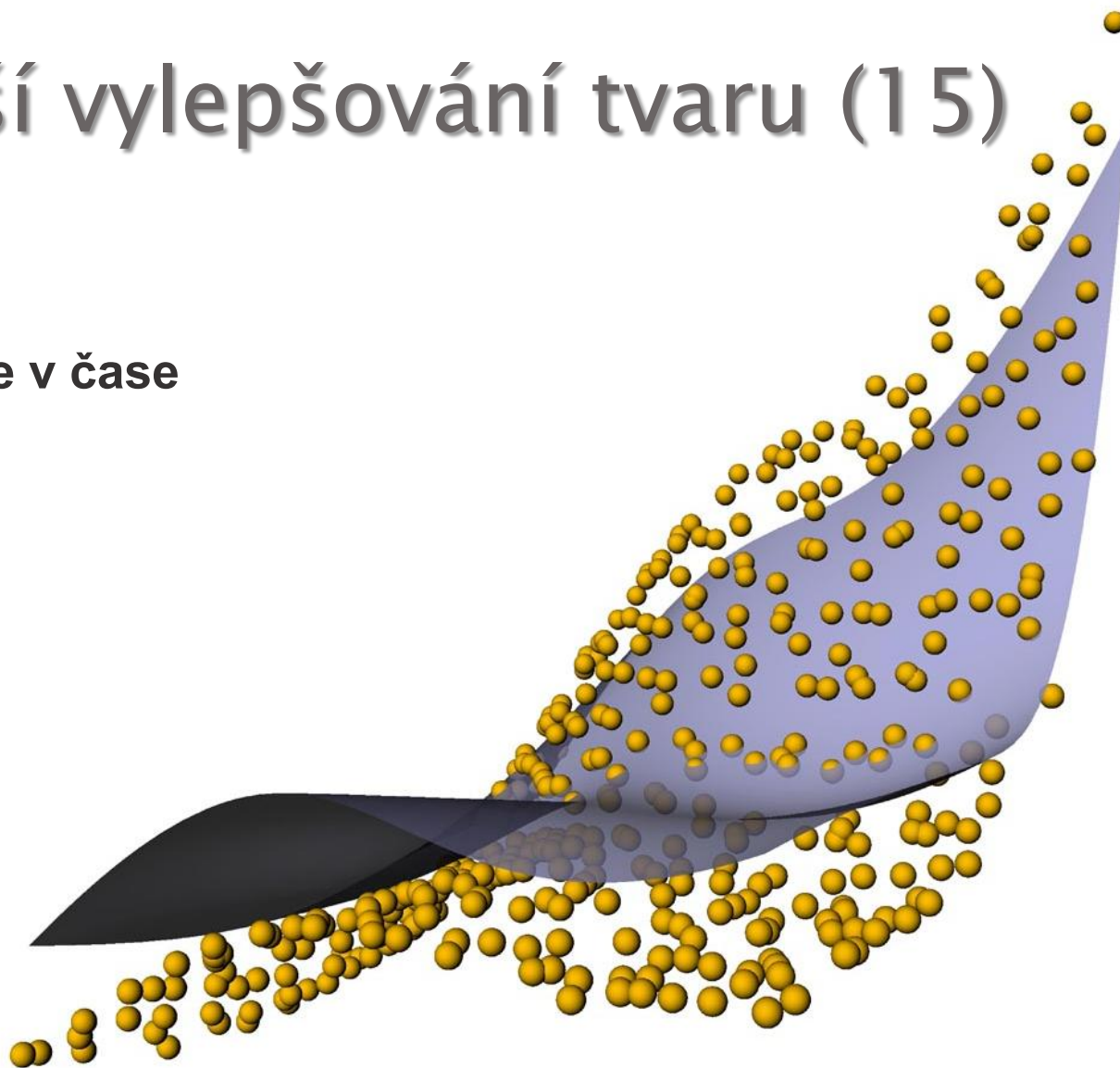
# Další vylepšování tvaru (14)

evoluce v čase



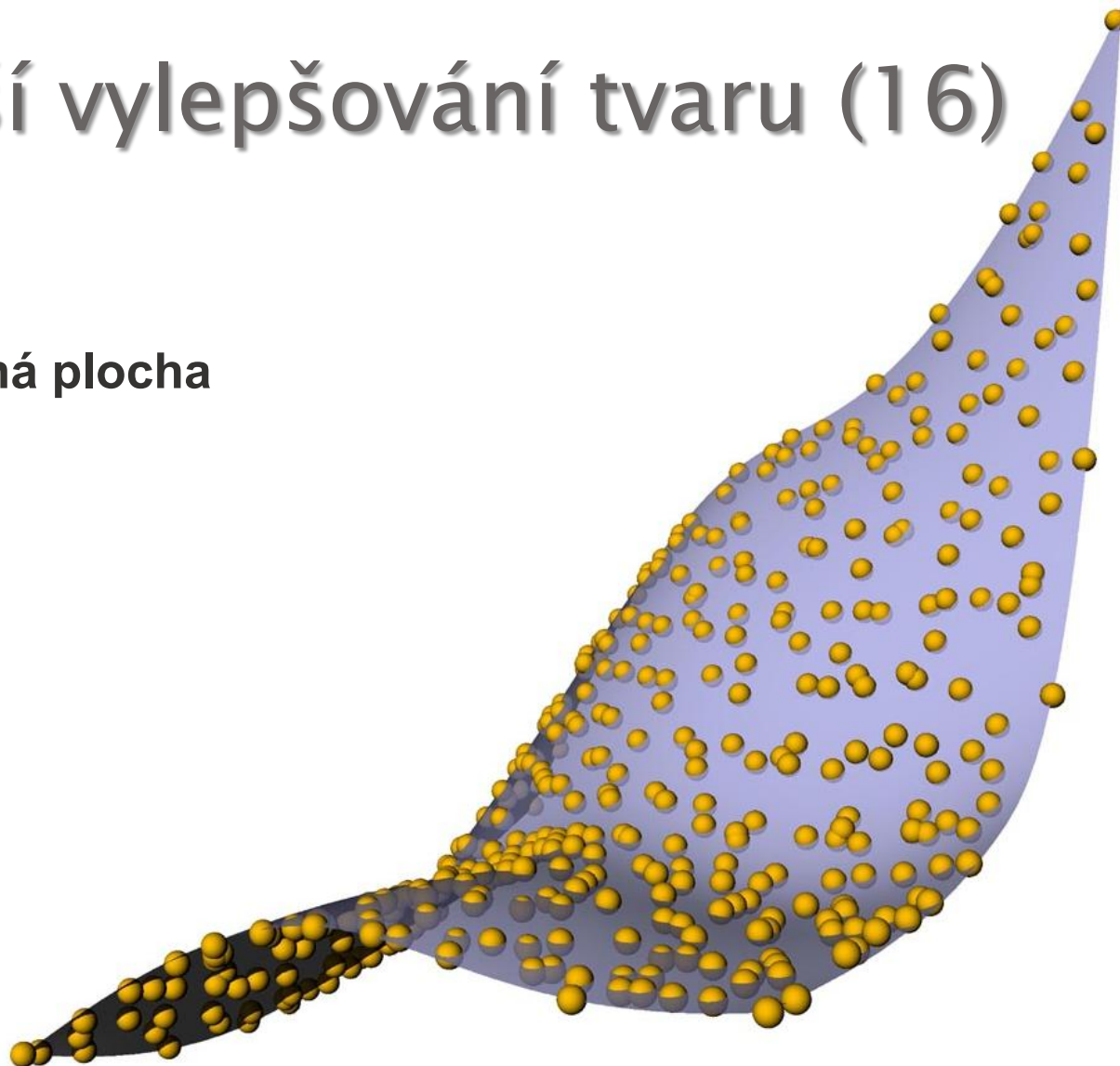
# Další vylepšování tvaru (15)

evoluce v čase



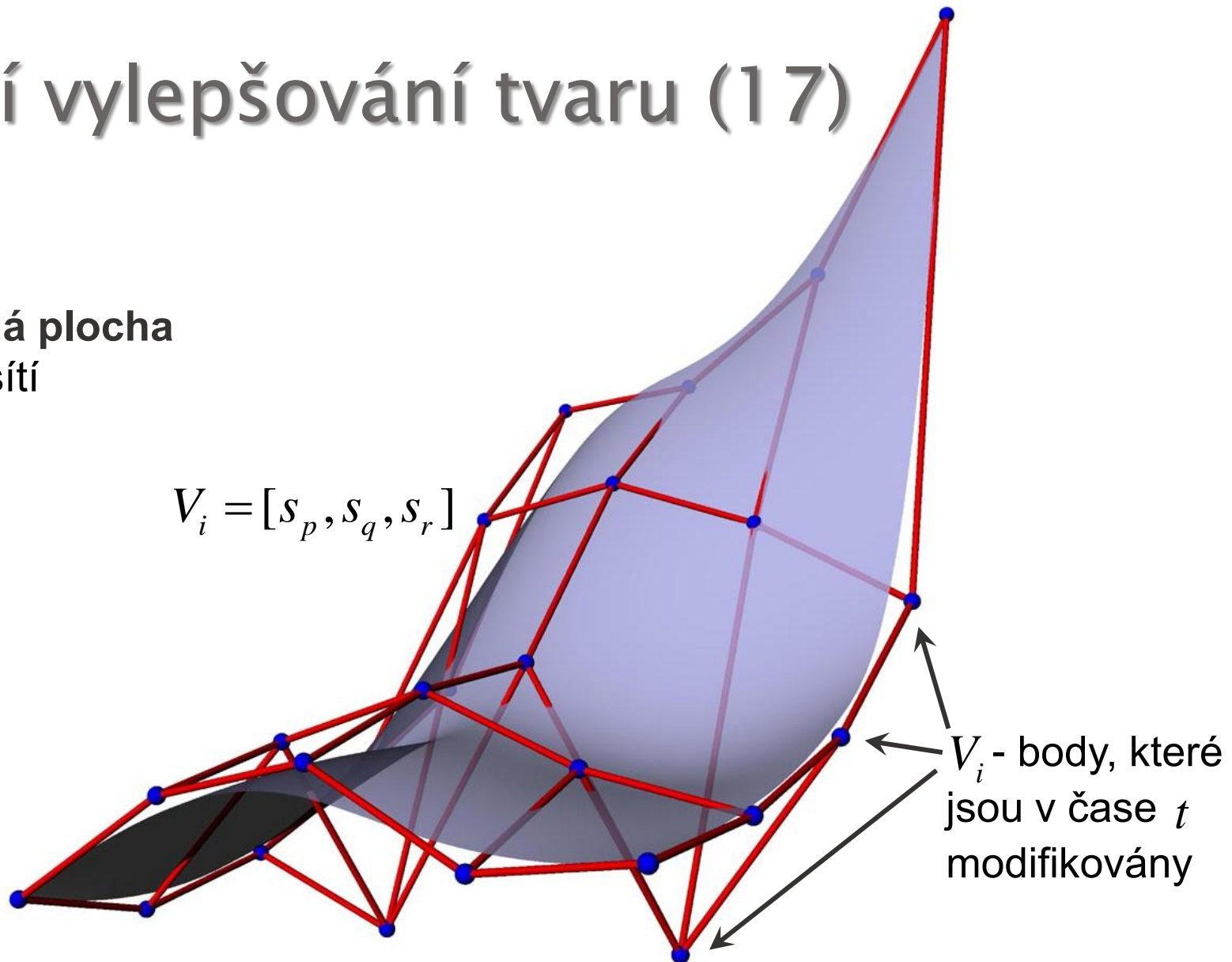
# Další vylepšování tvaru (16)

výsledná plocha



# Další vylepšování tvaru (17)

výsledná plocha  
s řídicí sítí



# Další vylepšování tvaru (18)

- Princip postupného vývoje tvaru je **analogický pro plochy**
  - jde ale o mnohem komplikovanější problém
  - to je také důvod, proč počáteční polohu a tvar plochy nevolíme zcela libovolně, ale **vycházíme z prvotní aproximace**
- Aproximace je sama o sobě velmi spolehlivou metodou
  - navíc ji lze využít identifikaci nesprávných bodů ve vstupní množině
  - protože vycházíme z aproximace metodou nejmenších čtverců, **nelze již plochu v tomto směru vylepšovat**
    - **plocha v lokálním minimu**
  - volíme **jiné funkční závislosti**
    - vylepšujeme tvar plochy tak, aby lépe aproximovala danou množinu bodů

# Další vylepšování tvaru (19)

- K daným bodům množiny se plochou přibližujeme minimalizací **chybové funkce**
  - chybová funkce např. vyjadřuje součet vzdáleností bodů vstupní množiny od **nejbližších bodů na ploše**
  - lze použít i jiné funkce
    - např. radiální bázové funkce – spojitě, diferencovatelné
- Minimalizace **chybové funkce**
  - **výpočet parciálních derivací**
  - **aktualizace řídicích bodů plochy**

$$P_x \text{ _ akt} = P_x \text{ _ poc} - \varepsilon \frac{1}{\text{parc\_der\_x}}$$

# Další vylepšování tvaru (20)

- **Jiný přístup**
  - po aproximaci Bézierovou plochou přecházíme **k racionální parametrizaci**
- **Další tvar plochy vylepšujeme změnou vah řídicích bodů**
  - **minimalizujeme opět chybovou funkci**, ale tentokrát aktualizujeme váhy řídicích bodů
  - **řídicí body zůstávají na místě**
  - **nemění se stupeň plochy**
  - **modifikuje se tvar plochy**
- **Použití jak pro funkci vzdáleností tak pro radiální funkce**



# Další vylepšování tvaru (21)

- **Radiální funkce**

- **volba bodu** v mračnu bodů (ve vstupní množině)
- váha ostatních bodů v mračnu je vůči zvolenému bodu vyjádřena radiální funkcí
  - např.  $vaha(x, y, z) = 1 + r(\varphi)$ 
    - $\varphi$  - vyjadřuje vzdálenost bodu v mračnu od zvoleného bodu (eukl.)
    - $r$  - radiální funkce,  $N$  - počet bodů mračna

$$chybova\_funkce = \sum_{i=1}^N nej\_vz\_dal\_k\_plose(x_i, y_i, z_i) \cdot vaha(x_i, y_i, z_i)$$

- např.  $r(\varphi) = e^{-(\varepsilon\varphi)^2}$
- chybová funkce závisí na řídicích bodech nebo na jejich vahách

# Programová demonstrace



**Ukázka postupného vylepšování tvaru  
Bézierovy plochy v prostředí Matlab**

# Závěr a budoucí práce

- V budoucí práci se zaměříme na použití **dalších typů ploch k rekonstrukci povrchů**
  - **B-spline a NURBS plochy**
    - Bézierovy plochy – nevýhodné při změně řídicího bodu, změna celé plochy, proto použití racionálních Bézierových ploch
    - jejich implementace, použití k aproximaci
- **Další vylepšování tvaru plochy**
  - použití jiných chybových funkcí
    - další radiální bázové funkce
  - interaktivní přístup – **možnost vybrat, ke kterým bodům se přiblížíme**

# Závěr a budoucí práce

- **Další vylepšování tvaru plochy**
  - **přidání vah k bodům mračna** – tj. vybírání důležitých bodů
  - **po aproximaci přejít k racionální parametrizaci a k plátování**
    - racionální parametrizace - hotové
  - **zvýšení stupně plochy**
    - ne vždy výhodné, může vzniknout nežádoucí vlnění
- **Další zpracování dat z 3D skenerů**
- **Vizualizace výsledků v programu Rhinoceros**