

Názov prednášky:

Vybrané interpolačné metódy

Osnova prednášky:

- Metóda trendového povrchu
- Multivariačný splajn

Odporúčaná literatúra

KAŇUK, J., 2015: Priestorové analýzy a modelovanie. Vysokoškolské učebné texty. Prírodovedecká fakulta UPJŠ v Košiciach, 106 s.

NETELER, M., MITASOVA, H.: Open Source GIS: A GRASS GIS Approach. Second Edition. Boston: Kluwer Academic Publisher, 2004.

Metóda trendového povrchu

- patrí medzi aproximačné metódy
- aplikácia tejto metódy pre analýzu sledovaného javu predpokladá normálne rozdelenie jeho hodnôt
- nerešpektuje priestorovú autokoreláciu, iba sa snažia prechádzať čo najbližšie zadanými bodmi
- môže byť globálna aj lokálna
- trendová funkcia je vyjadrená polynomicou funkciou určitého rádu. Jej všeobecný zápis by sme mohli vyjadriť nasledovne:

$$T(m) = c_1 + c_2x + c_3x^2 + \dots + c_mx^{m-1} + c_{m+1}x^m$$

- trendový povrch určený polynómom vyššieho rádu však vykazuje značné chyby na okrajových pozíciách sledovaného javu (edge effects)
- pre dvojrozmerný priestor je potrebné do rovnice zakomponovať obe súradnice x a y
- cieľom tejto metódy je odhadnúť priebeh sledovaného javu pomocou polynomickej funkcie metódou najmenších štvorcov

Metóda najmenších štvorcov

- je navrhnutá tak, aby funkcia mala istú voľnosť, čo znamená, že nemusí prechádzať zadanými bodmi, avšak mala by prechádzať v ich blízkosti
- metóda bola navrhnutá pre odhad hodnôt dát získané experimentálnym meraním. Jednotlivé hodnoty môžu byť zaťažené chybou merania, ktoré vznikli v dôsledku metódy merania. Ak by sme chyby nezohľadňovali, výsledný povrch by kopíroval chybu merania, a preto by bol nereálny
- najjednoduchším prípadom je lineárna závislosť, ktorej výsledkom je priamka.
- pre vstupné bodové pole sa snažíme nájsť takú lineárnu funkciu, ktorá bude prechádzať bodmi vstupného bodového poľa čo najbližšie k nim. Vychádzajme preto so situácie, že sú nám známe body x_i , $i = 1, 2, \dots, n$ a funkčné hodnoty v nich sú y_i . Budeme teda hľadať priamku, pre ktorú je možné vyjadriť vzťahom

$$y = c_1 + c_2x$$

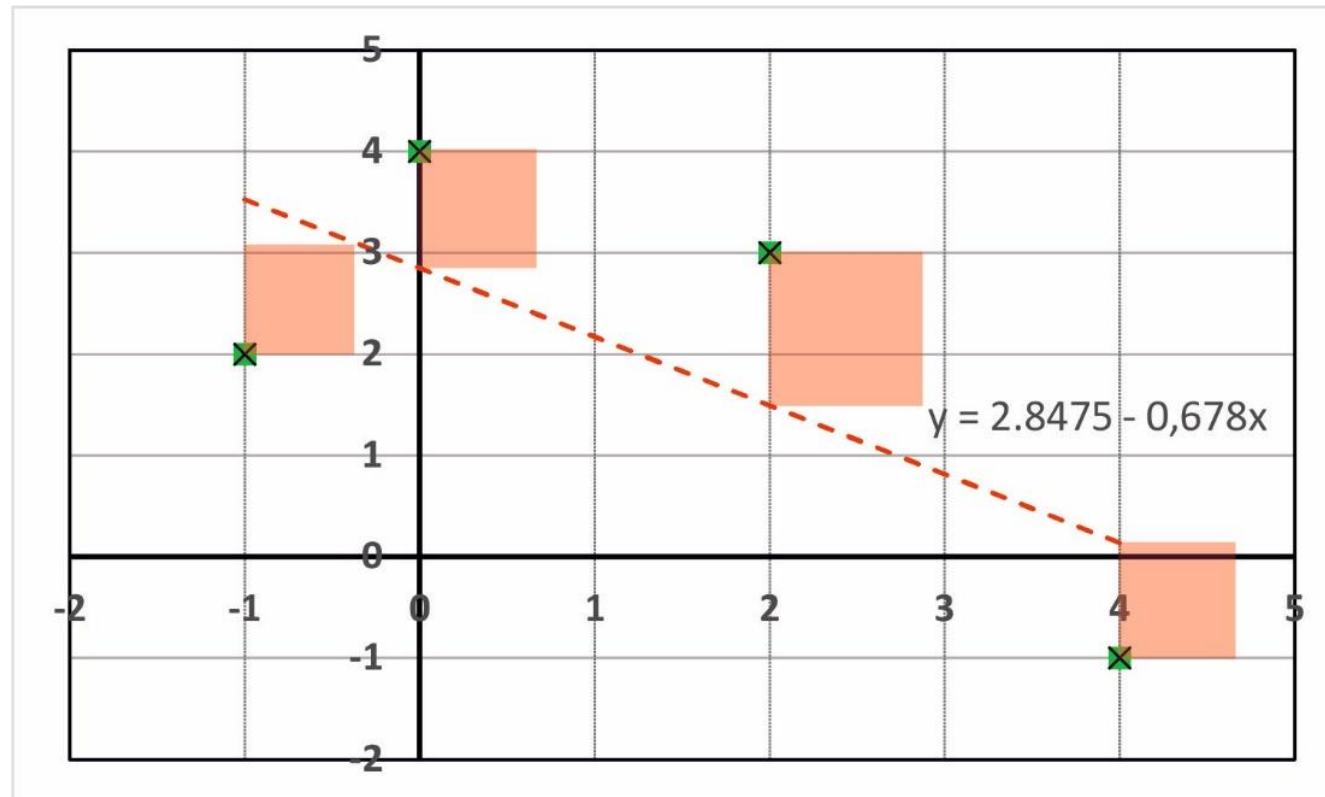
- priamka má prechádzať zadanými bodmi v najbližšej možnej vzdialenosti. Preto si vyjadríme chybu odhadu e_i , pre ktorú platí vzťah:

$$e_i = y_i - y(x_i) = y_i - c_1 - c_2x_i$$

Metóda najmenších štvorcov

- najlepším kritériom pre dosiahnutie najbližšej možnej vzdialenosti je, aby súčet štvorcov $S(c_1, c_2)$ (druhých mocnín chýb) bol minimálny, čo môžeme zapísať nasledovne:

$$\begin{aligned} S(c_1, c_2) &= (y_1 - c_1 - c_2 x_1)^2 + (y_2 - c_1 - c_2 x_2)^2 + \dots + (y_n - c_1 - c_2 x_n)^2 = \\ &= \sum_{i=1}^n (y_i - c_1 - c_2 x_i)^2 = \min S(c_1, c_2) \end{aligned}$$



Metóda najmenších štvorcov

- Ostré lokálne minimum funkcie $S(c_1, c_2)$ môže nastať v bode (c_1, c_2) , v ktorom sú nulové parciálne derivácie, čiže platí:

$$\frac{\partial S}{\partial c_1} = 0 \qquad \frac{\partial S}{\partial c_2} = 0$$

Po dosadení a výpočte parciálnej derivácie pre c_1 a c_2 dostaneme sústavu rovníc:

$$c_1(n) + c_2 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$c_1 \sum_{i=1}^n x_i + c_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

Metóda trendového povrchu

- príklad

i	1	2	3	4
x_i	-1	0	2	4
y_i	2	4	3	-1

Najskôr vypočítame potrebné koeficienty

i	x_i	y_i	x_i^2	$x_i y_i$
1	-1	2	1	-2
2	0	4	0	0
3	2	3	4	6
4	4	-1	16	-4
Σ	5	8	21	0

Po dosadení do sústavy rovníc

$$c_1(n) + c_2 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$c_1 \sum_{i=1}^n x_i + c_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

dostávame tvar rovníc nasledovne:

$$4c_1 + 5c_2 = 8$$

$$5c_1 + 21c_2 = 0$$

Riešením týchto rovníc je

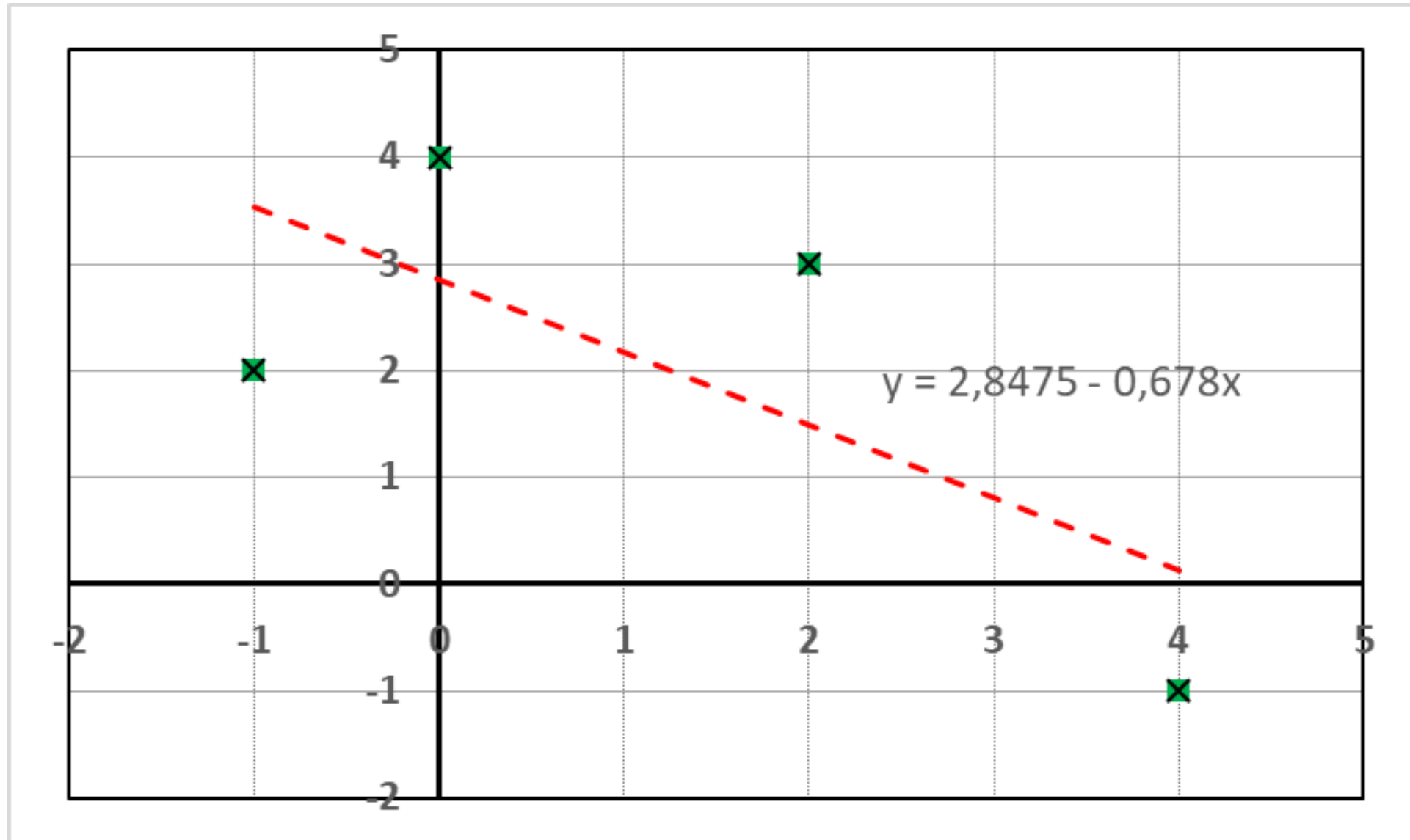
$$c_1 = 2,8475$$

$$c_2 = -0,678$$

Hľadanú priamku vyjadríme funkciou $y = c_1 + c_2 x$

$$y = 2,8475 - 0,678x$$

Metóda najmenších štvorcov



Metóda najmenších štvorcov

V prípade, že chceme hľadať priebeh funkcie pre body pomocou polynómov vyššieho rádu, postup bude analogický, ako v prípade priamky, resp. polynomickej funkcie prvého rádu. Všeobecný zápis pre polynomicnú funkciu m rádu je:

$$S(m) = c_1 + c_2x + c_3x^2 + \dots + c_mx^{m-1} + c_{m+1}x^m$$

Pričom by sme podobne ako u priamky dostali sústavu rovníc podľa rádu polynomickej funkcie, ktorých všeobecný zápis vyzerá nasledovne:

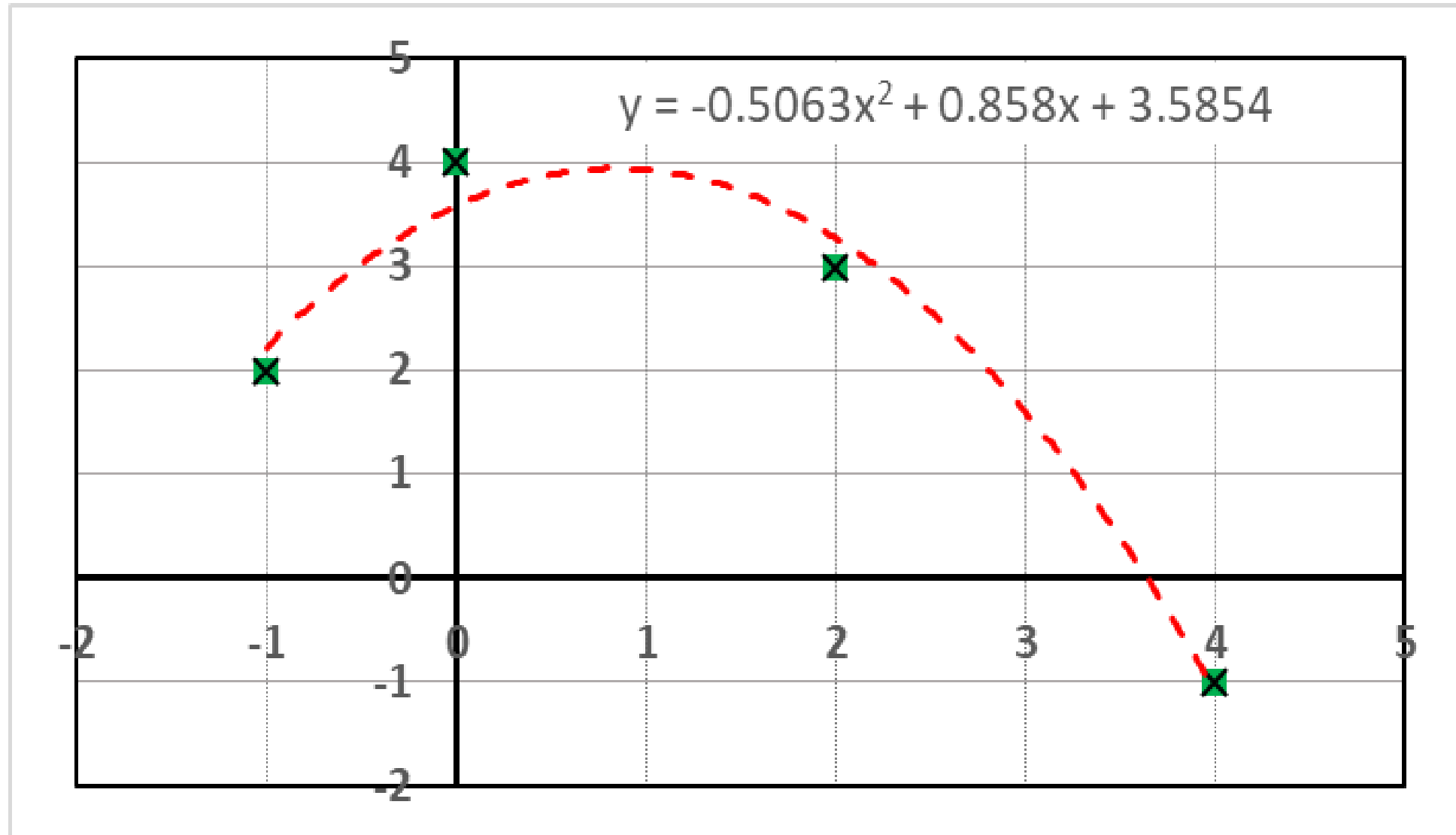
$$c_1(n) + c_2 \sum_{i=1}^n x_i + \dots + c_m \sum_{i=1}^n x_i^{m-1} = \sum_{i=1}^n y_i$$

$$c_1 \sum_{i=1}^n x_i + c_2 \sum_{i=1}^n x_i^2 + \dots + c_m \sum_{i=1}^n x_i^m = \sum_{i=1}^n x_i y_i$$

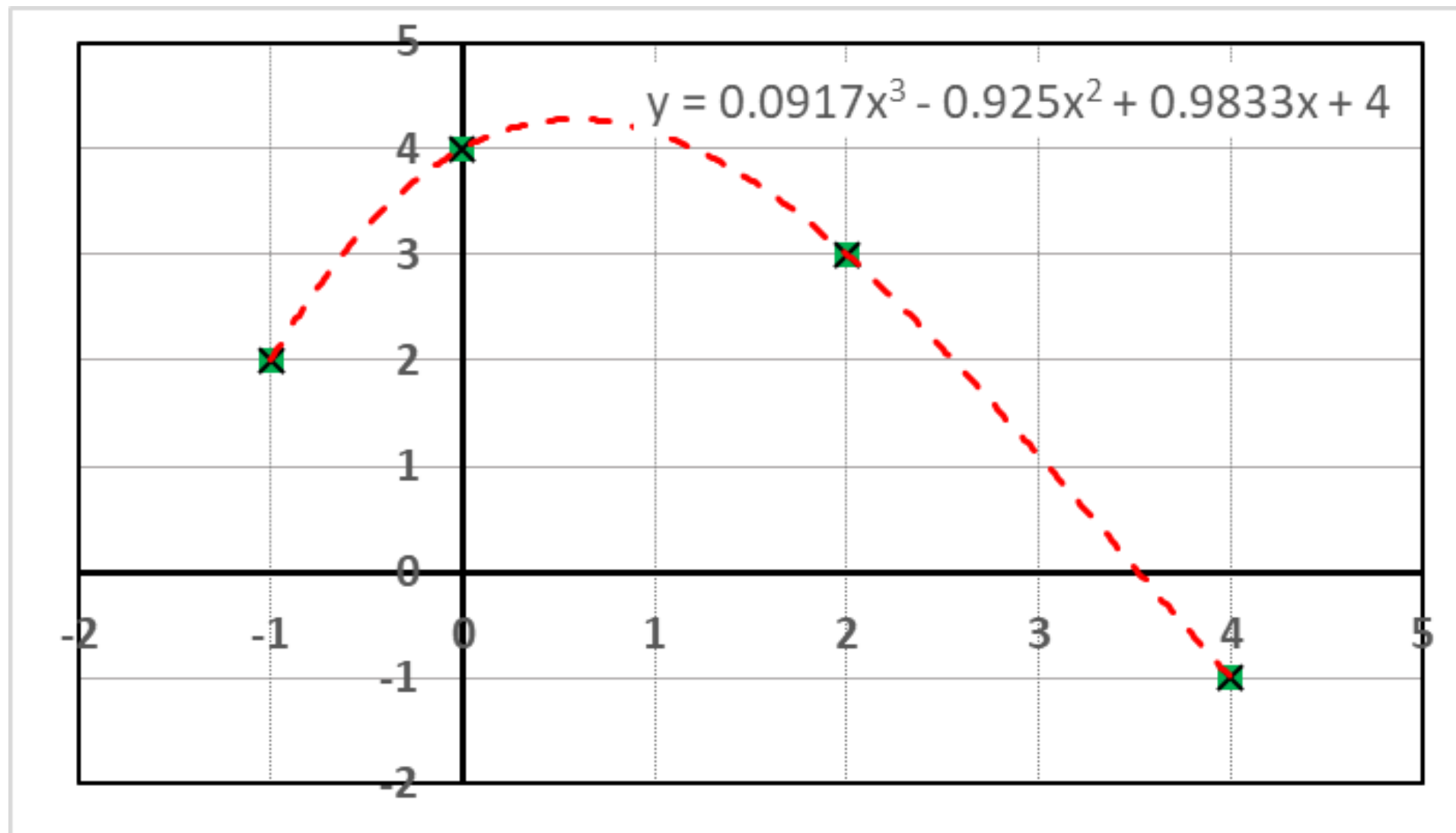
$$c_1 \sum_{i=1}^n x_i^2 + c_2 \sum_{i=1}^n x_i^3 + \dots + c_m \sum_{i=1}^n x_i^{m+1} = \sum_{i=1}^n x_i^2 y_i$$

$$c_1 \sum_{i=1}^n x_i^m + c_2 \sum_{i=1}^n x_i^{m+1} + \dots + c_m \sum_{i=1}^n x_i^{2m} = \sum_{i=1}^n x_i^m y_i$$

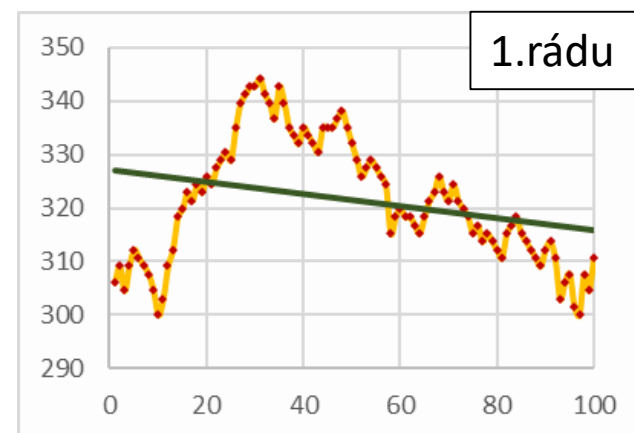
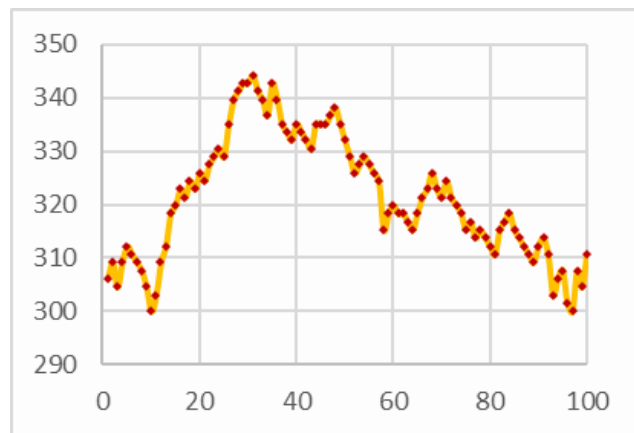
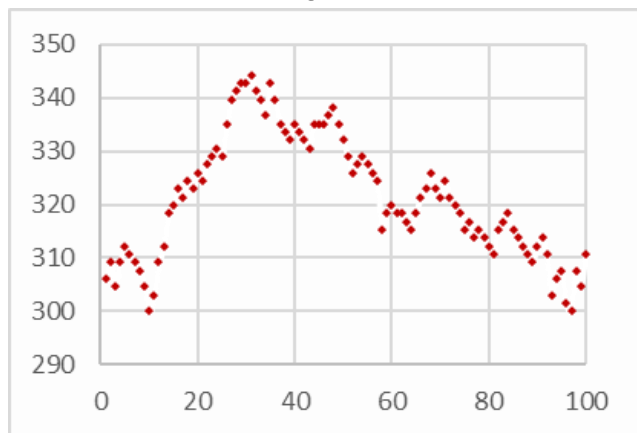
Metóda najmenších štvorcov



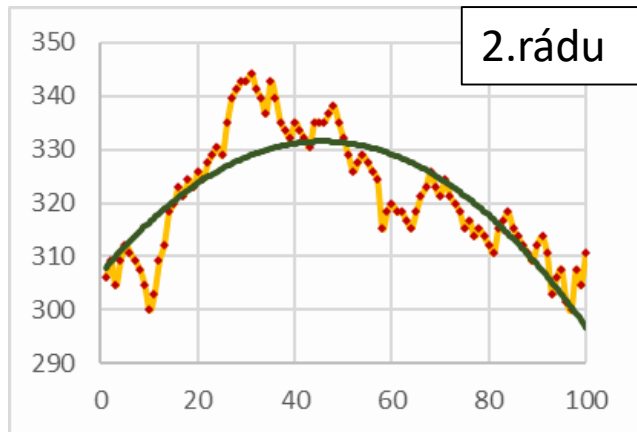
Metóda najmenších štvorcov



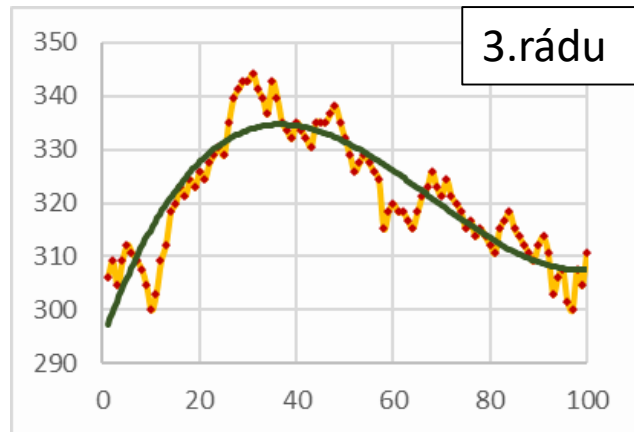
Metóda najmenších štvorcov



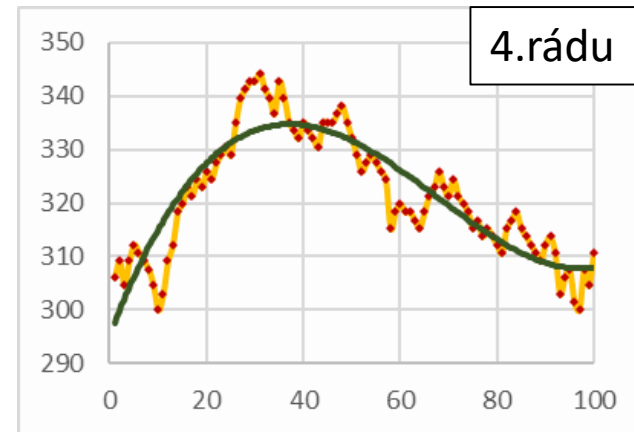
1.rádu



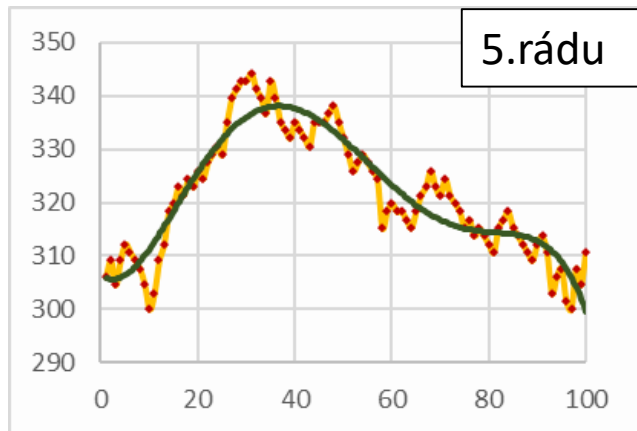
2.rádu



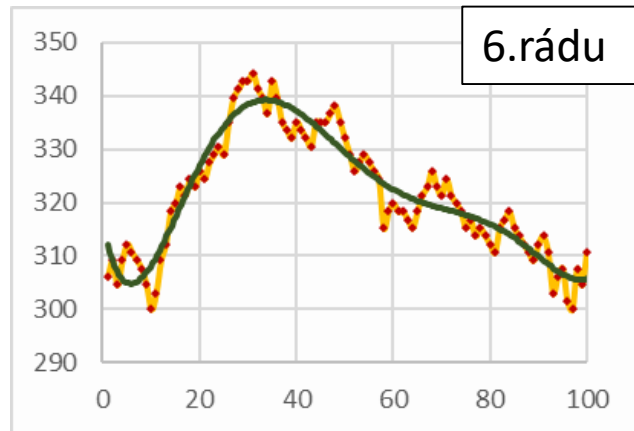
3.rádu



4.rádu



5.rádu



6.rádu

$$y = -0,1125x + 327,07$$

$$y = -0,0118x^2 + 1,0813x + 306,77$$

$$y = 0,0002x^3 - 0,0465x^2 + 2,4903x + 294,62$$

$$y = 0,0000003x^4 + 0,0002x^3 - 0,0421x^2 + 2,3899x + 295,15$$

$$y = -0,0000002x^5 + 0,0000624x^4 - 0,0054x^3 + 0,1712x^2 - 0,7499x + 306,46$$

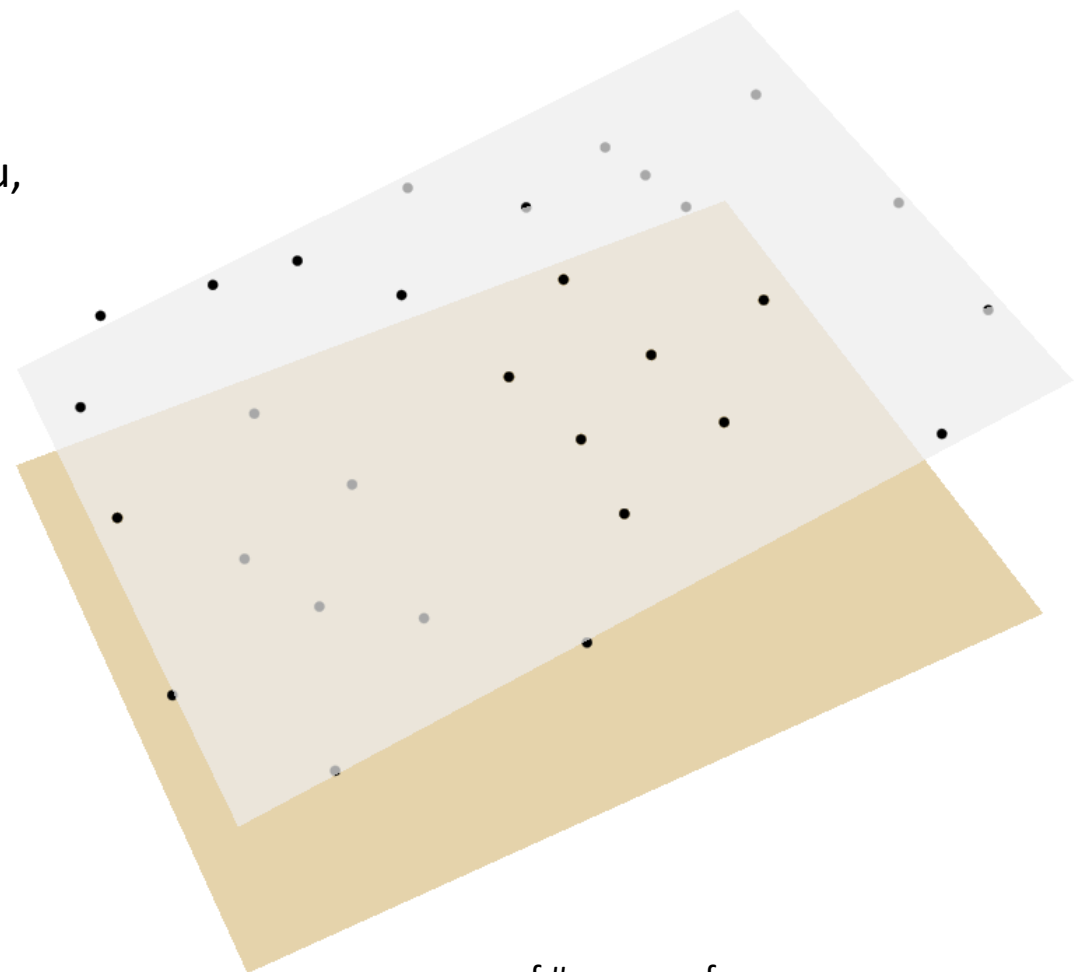
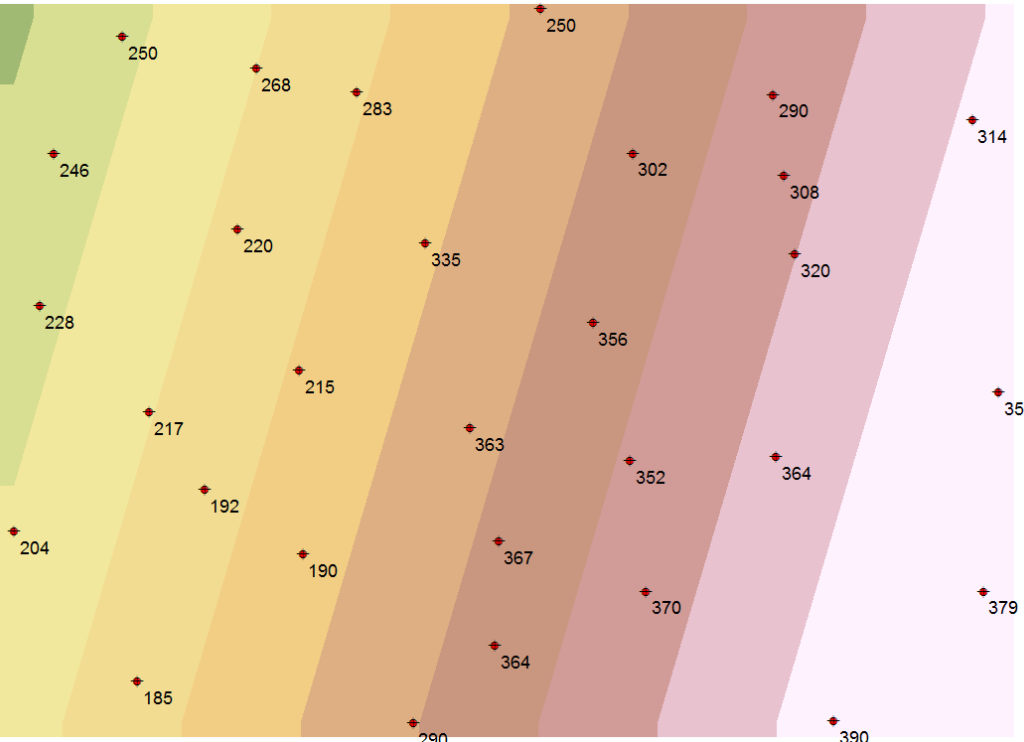
$$y = 0,00000001x^6 - 0,00000241x^5 + 0,00031x^4 - 0,019x^3 + 0,515x^2 - 4,332x + 315,93$$

Metóda trendového povrchu

- najjednoduchšia forma polynomickej funkcie je funkcia prvého rádu, ktorú označujeme ako lineárny trend $T(1)$ a môžeme ju vyjadriť nasledovne

$$T(1) = c_1 + c_2x + c_3y$$

c_1, c_2, c_3 sú koeficienty chýb

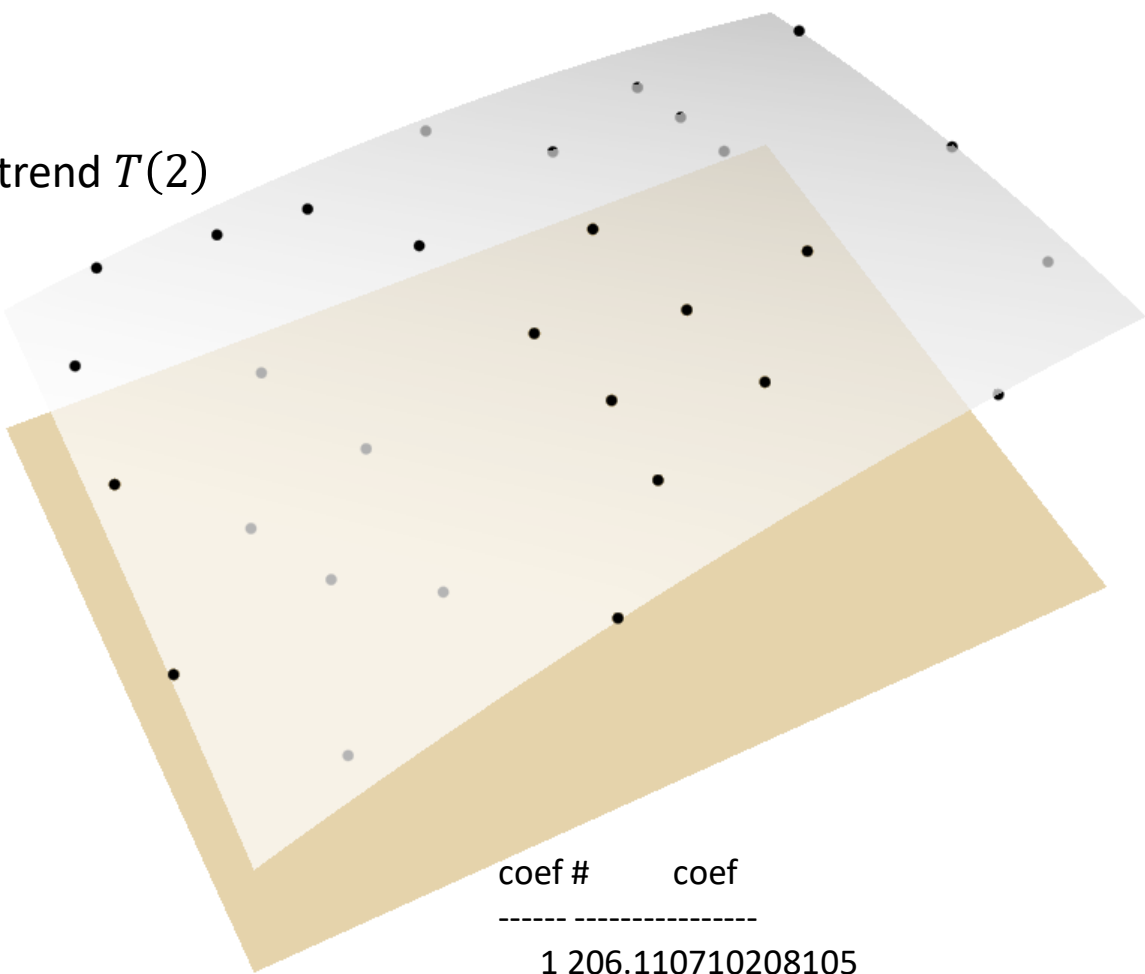


coef #	coef
1	181.754599966162
2	0.0947529006546955
3	-0.0280793391721109
RMS Error = 40.6461319654569	

Metóda trendového povrchu

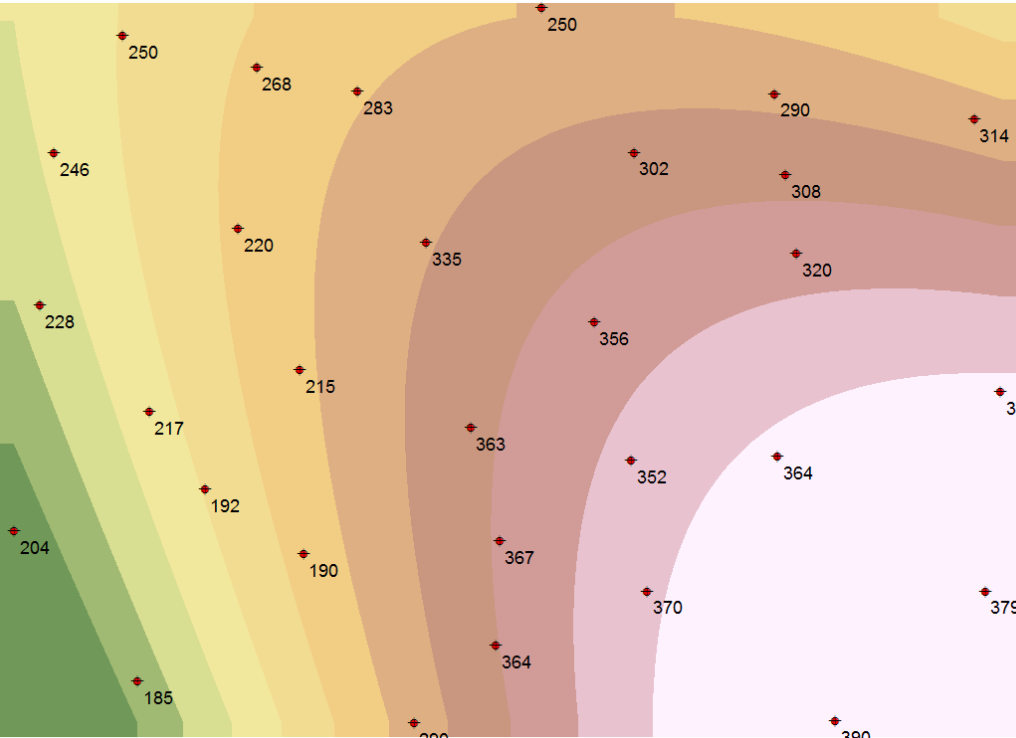
- polynomická funkcia druhého rádu sa označuje ako kvadratický trend $T(2)$

$$T(2) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2$$



coef #	coef
1	206.110710208105
2	0.121950703599808
3	0.0219544979199638
4	-5.74442183891761e-005
5	-0.000125116275482547
6	-5.04551201767515e-005

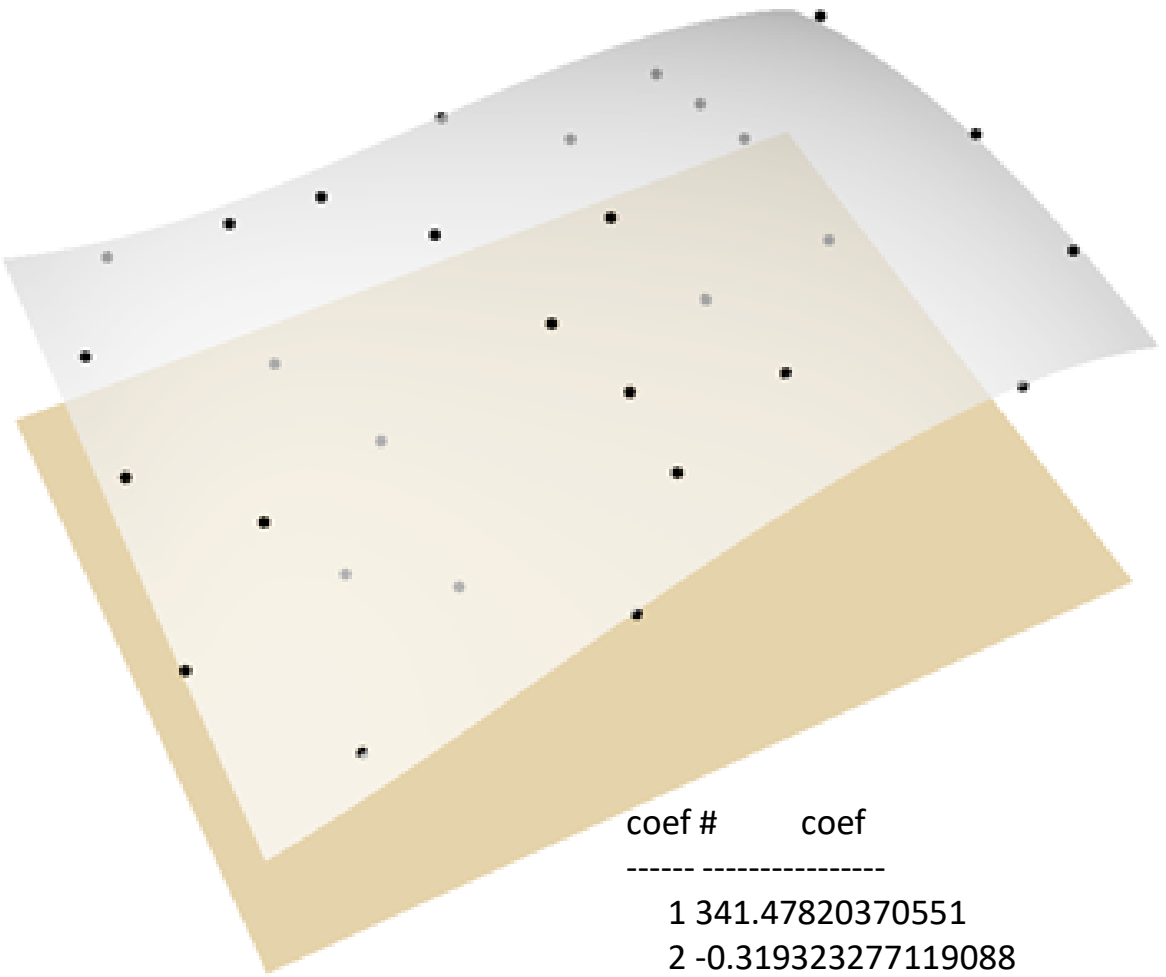
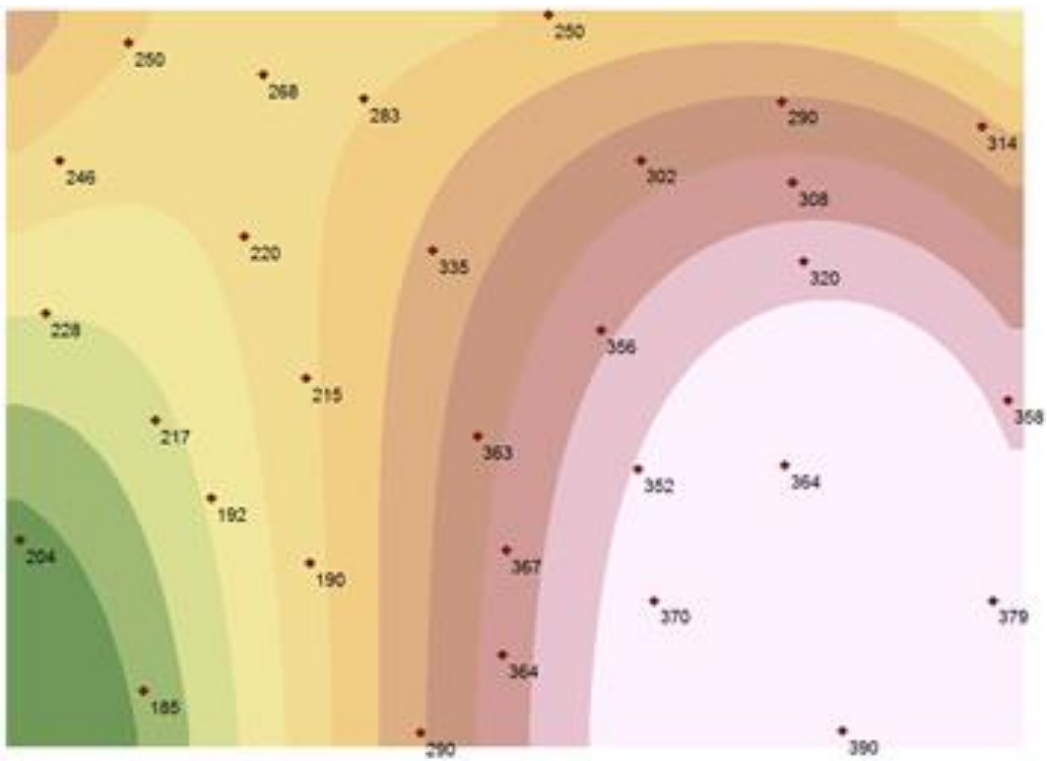
RMS Error = 30.5856426818541



Metóda trendového povrchu

- Polynomickú funkciu tretieho rádu nazývame kubický trend $T(3)$

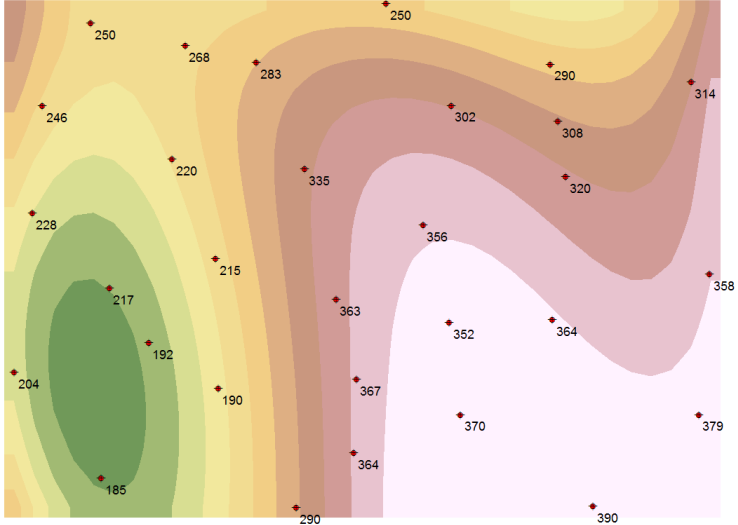
$$T(3) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$$



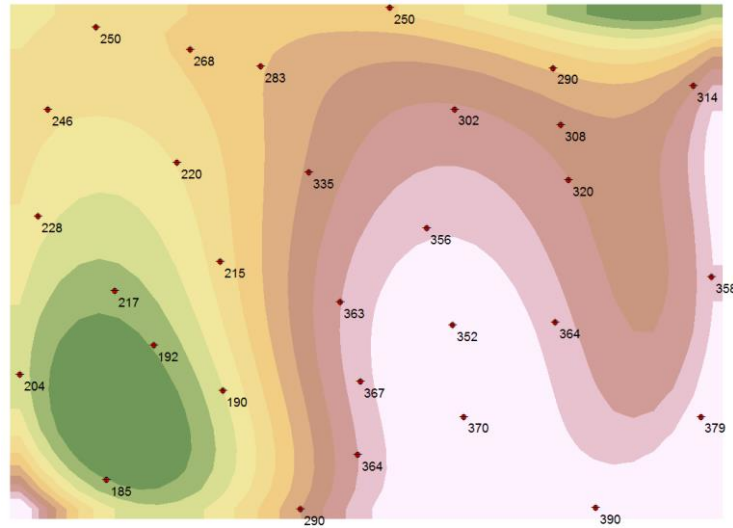
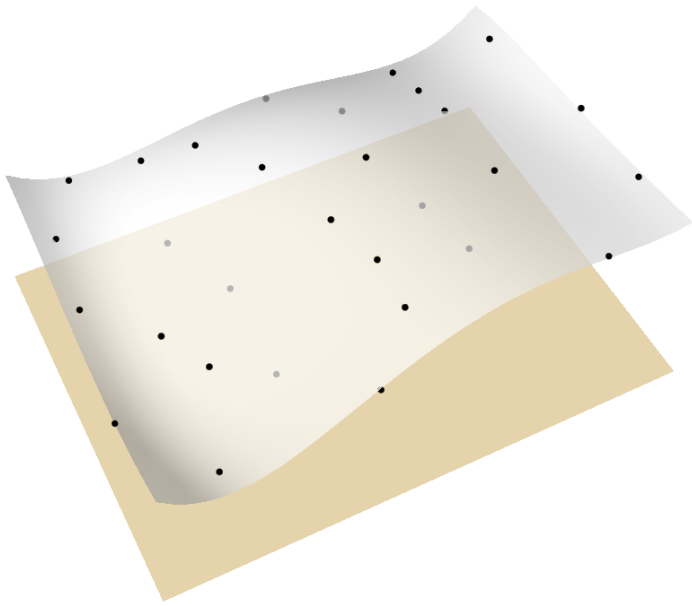
coef #	coef
1	341.47820370551
2	-0.319323277119088
3	0.19579846667767
4	0.000333108736456384
5	-0.000439066527275171
6	-1.75998721882544e-007
7	-1.07814861016934e-007
8	8.62508330285841e-008
9	-9.5392719504918e-008
10	-1.97367544209482e-008

RMS Error = 26.5271644980927	

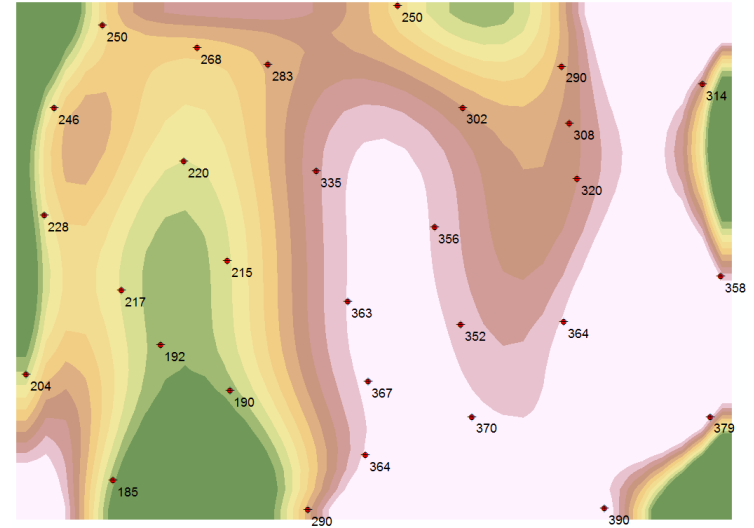
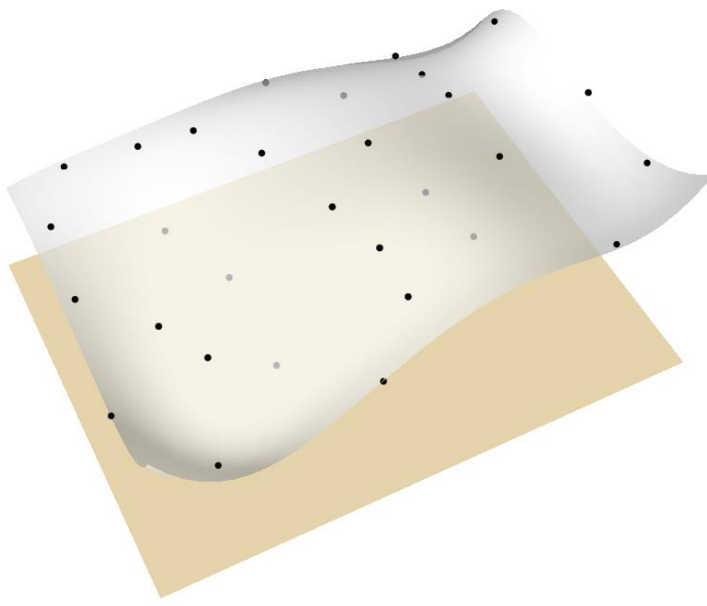
Metóda trendového povrchu



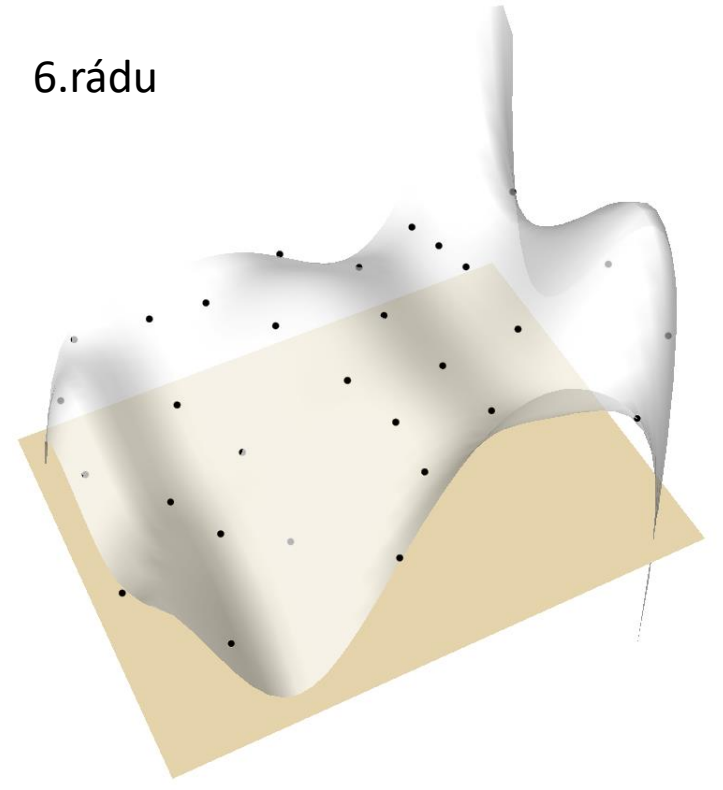
4.rádu



5.rádu



6.rádu



Metóda trendového povrchu

- z uvedených príkladov vyplýva, že síce RMS klesá s vyšším rádom polynomickej funkcie
- funkcia veľmi kmitá hlavne na okrajoch bodového poľa, kde aproximácia hodnôt je nereálna.
- na druhej strane v strede bodového poľa môžeme pozorovať, že pomocou polynomickej funkcie vyššieho rádu dostávame veľmi hladký povrch.
- pri polynomickej funkcii šiesteho rádu je fluktuácia hodnôt bodového poľa už veľmi veľká v celom rozsahu dát, aj keď RMS je najnižšia, nakoľko funkcia prechádza v blízkosti bodov vstupného bodového poľa.
- interpretácia aproximovaného povrchu trendovými polynomickými funkciami vyššieho rádu je veľmi zložitá.
 - pri nižších rádoch ešte môžeme interpretovať tendenciu sledovaného javu z hľadiska priebehu funkcie vo všetkých smeroch, t.j. či povrch má tendenciu stúpať, alebo klesať.
 - zároveň s vyšším rádom polynomickej funkcie stúpa aj náročnosť výpočtu, nakoľko rastie počet premenných, a tým aj počet sústavy rovníc.
- ak chceme metódu trendového povrchu aplikovať pre odhad priebehu hodnôt bodového poľa, je vhodnejšie používať lokálny prístup, nakoľko pri použití jednej funkcie na celý rozsah vstupného bodového poľa (globálny prístup) dostávame výsledky nezlučiteľné s reálnym priebehom hodnôt (pri vysokom kmitaní funkcie dostávame nereálne výsledky, napr. pri modelovaní teploty by sme mohli dostať aj záporné hodnoty, podobne aj v prípade nadmorskej výšky).

Interpolačný polynóm

- interpolačný polynóm sa využíva pre interpoláciu malého počtu hodnôt, ktoré nie sú zaťažené chybou merania, pričom chceme dosiahnuť, aby výsledná funkcia prechádzala danými bodmi. Podobne je to aj v prípade splajnov, ktoré sú navrhnuté pre väčší počet bodov.
- Pre zápis vyjadrenia priebehu funkcie sa pokúsme vychádzať z toho, že hľadáme funkciu pre určenie bodu zo súradnicami $[x_0, y_0]$, kde platí, že:

$$y_0 = f(x_0)$$

$$f(x_0) = P_{n+1}(x_0)$$

- pričom počet bodov vstupného bodového poľa $n > 2$ a ktorý prechádza bodmi zo súradnicami $[x_1, y_1], [x_2, y_2], \dots, [x_{n+1}, y_{n+1}]$, pričom platí, že:

$$x_1 < x_2 < \dots < x_n < x_{n+1}$$

- hodnotu y_0 v bode x_0 určíme na základe interpolačného polynómu $P_{n+1}(x_0)$, ktorého hodnotu vyjadríme pomocou kombinácií funkcií $f_{n+1}(x_0)$, kde platí, že:

$$f_i(x_j) = \begin{cases} 1 & \text{pre } i = j \\ 0 & \text{pre } i \neq j \end{cases}$$

Interpolačný polynóm

- Potom platí, že:

$$f_1(x_0) = \frac{(x_0 - x_2)(x_0 - x_3) \dots (x_0 - x_{n+1})}{(x_1 - x_2)(x_1 - x_3) \dots (x_1 - x_{n+1})}$$

$$f_2(x_0) = \frac{(x_0 - x_1)(x_0 - x_3) \dots (x_0 - x_{n+1})}{(x_2 - x_1)(x_2 - x_3) \dots (x_2 - x_{n+1})}$$

$$f_i(x_0) = \frac{(x_0 - x_1)(x_0 - x_{i-1})(x_0 - x_{i+1}) \dots (x_0 - x_{n+1})}{(x_i - x_1)(x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_{n+1})}$$

$$f_{n+1}(x_0) = \frac{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)}{(x_{n+1} - x_1)(x_{n+1} - x_2) \dots (x_{n+1} - x_n)}$$

- interpolačný polynóm $P_{n+1}(x_0)$, následne dostaneme ako kombináciu funkcií $f_{n+1}(x_0)$ pre súradnice v bodoch y_1 až y_{n+1} nasledovne:

$$P_{n+1}(x_0) = y_1 f_1(x_0) + y_2 f_2(x_0) + \dots + y_{n+1} f_{n+1}(x_0)$$

Interpolačný polynóm

- čiže po dosadení vyzerá zápis polynómu nasledovne:

$$\begin{aligned} P_{n+1}(y_0) &= \\ &= y_1 \frac{(x_0 - x_2)(x_0 - x_3) \dots (x_0 - x_{n+1})}{(x_1 - x_2)(x_1 - x_3) \dots (x_1 - x_{n+1})} + y_2 \frac{(x_0 - x_1)(x_0 - x_3) \dots (x_0 - x_{n+1})}{(x_2 - x_1)(x_2 - x_3) \dots (x_2 - x_{n+1})} + \dots \\ &+ y_{n+1} \frac{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)}{(x_{n+1} - x_1)(x_{n+1} - x_2) \dots (x_{n+1} - x_n)} \end{aligned}$$

- uvedená funkcia sa nazýva Lagrangeov interpolačný polynóm ($P_{n+1}(y_0)$)

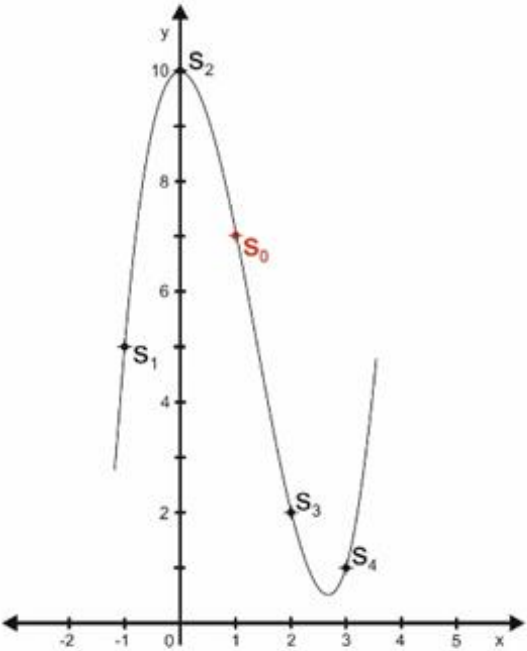
Interpolačný polynóm

- Lagrangeov interpolačný polynóm - príklad

i	1	2	3	4
x	-1	0	2	3
y	?	?	?	?

$$\begin{aligned} P_{n+1}(x_0) &= 5 \frac{(x_0 - 0)(x_0 - 2)(x_0 - 3)}{(-1 - 0)(-1 - 2)(-1 - 3)} + 10 \frac{(x_0 - (-1))(x_0 - 2)(x_0 - 3)}{(0 - (-1))(0 - 2)(0 - 3)} + 2 \frac{(x_0 - (-1))(x_0 - 0)(x_0 - 3)}{(2 - (-1))(2 - 0)(2 - 3)} \\ &+ 1 \frac{(x_0 - (-1))(x_0 - 0)(x_0 - 2)}{(3 - (-1))(3 - 0)(3 - 2)} = x^3 - 4x^2 + 10 \end{aligned}$$

i	1	2	3	4
x	-1	0	2	3
y	5	10	2	1

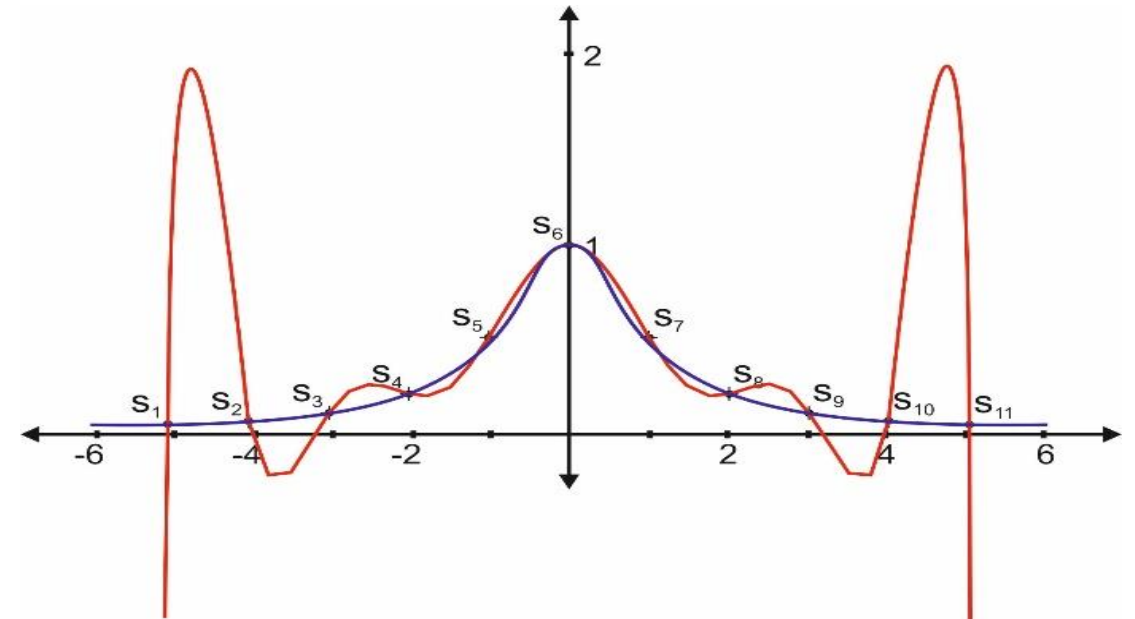


Interpolačný polynóm

- Lagrangeov interpolačný polynóm – zhrnutie
 - je nepraktický v tom, že ak chceme pridať ďalší vstupný bod, celý polynóm musíme prepočítavať znova.
- existuje viacero postupov pre odhad funkcie $y_0 = f(x_0)$ napr. Newtonov interpolačný polynóm, ktorý pri výpočte určuje koeficienty pomocou tzv. pomerných diferencií.
- že výsledný polynóm je rovnaký bez ohľadu na použitý postup.
- dôležité je najmä to, aby sme polynomicke funkciu vedeli nájsť čo najrýchlejšie.

V prípade väčšieho počtu vstupných bodov však dochádza ku situáciám, kedy polynomicke funkcia prechádza zadanými bodmi, ale mimo nich sa veľmi rozkmitá.

Preto napríklad už pri desiatich bodoch sa interpolačný polynóm nehodí pre odhad hodnôt a je potrebné hľadať iné metódy. Riešenie je v použití splajnov.



Splajn

- charakterizujeme ich ako sústavu polynómov nižšieho stupňa na rôznych intervaloch, ktoré na seba navzájom nadväzujú v bodoch vstupného bodového poľa.
- Fajmon a Ružičková (2005) uvádza charakteristiky pre niekoľko radov polynomických funkcií:
 - a) **Lineárny splajn** (1 - rád) predstavuje sústavu lineárnych funkcií, ktoré na seba nadväzujú v zadaných bodoch. Ide vlastne o lomenú čiaru spájajúcu body vstupného bodového poľa. J
 - jej nevýhodou je to, že funkcia má veľmi ostré hrany v zadaných bodoch.
 - nie je možné určiť deriváciu v zadaných bodoch.
 - zároveň lineárny splajn môže medzi zadanými bodmi príliš oscilovať od skutočného povrchu (analogicky ako TIN).
 - b) **Kvadratický splajn** (2 - rád) by sme mohli charakterizovať ako sústavu kvadratických funkcií, ktoré na seba v zadaných bodov nadväzujú funkčnou hodnotou ako aj hodnotou prvej derivácie.
 - aj keď má lepšie vlastnosti ako lineárny splajn, jeho negatívnou vlastnosťou je jeho zotrvačnosť, nakoľko každé dva susedné body sú spojené parabolou, ktorá „prestrelí“ bod $[x_i, y_i]$, kým naberie „správny“ smer k bodu $[x_{i+1}, y_{i+1}]$. (obr. 4.2)
 - a) **Kubický splajn** (3 - rád) je sústava kubických funkcií, ktoré na seba nadväzujú funkčnou hodnotou, ako aj prvou a druhou deriváciou.
 - veľmi dobré vlastnosti, pretože prechádza zadanými bodmi a vystihuje priebeh lepšie ako parabola
 - v každom bode existuje prvá a druhá derivácia
- okrem lineárneho, kvadratického a kubického splajnu existujú aj ďalšie metódy, napr. logaritmická metóda

Splajn – kubický splajn

- prirodzenú kubickú funkciu $k(x_0)$ by sme mohli zapísať vo všeobecnom tvare

$$k(x_0) = a_i + b_i(x_0 - x_i) + c_i(x_0 - x_i)^2 + d_i(x_0 - x_i)^3$$

- Podmienky:

- kubickú funkciu budeme prekladať cez susedné body vstupného bodového poľa podľa intervalov $\langle x_i; x_{i+1} \rangle$, pričom $i = 1, 2, 3, \dots, n + 1$

- polynómy prechádzajú zadanými bodmi;

$$k_n(x_{n+1}) = f(x_{n+1})$$

- polynómy na seba nadväzujú

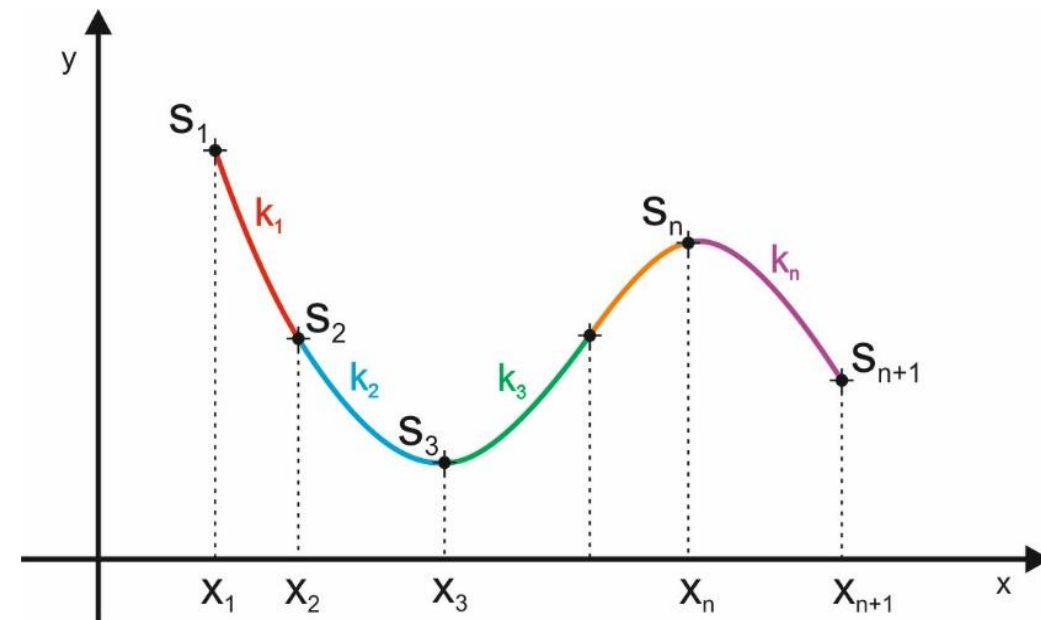
$$k_i(x_{i+1}) = k_{i+1}(x_{i+1}), \quad i = 1, 2, \dots, n - 1$$

- prvá derivácia polynómov na seba nadväzuje

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

- druhá derivácia polynómov na seba nadväzuje

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



Splajn – príklad

- prirodzenú kubickú funkciu $k(x_0)$ by sme mohli zapísať vo všeobecnom tvare

i	1	2	3	4
x_i	-1	0	2	4
y_i				

$$k_1(x_0) = a_1 + b_1(x_0 - x_1) + c_1(x_0 - x_1)^2 + d_1(x_0 - x_1)^3$$

$$x_0 \in \langle x_1; x_2 \rangle$$

$$k_2(x_0) = a_2 + b_2(x_0 - x_2) + c_2(x_0 - x_2)^2 + d_2(x_0 - x_2)^3$$

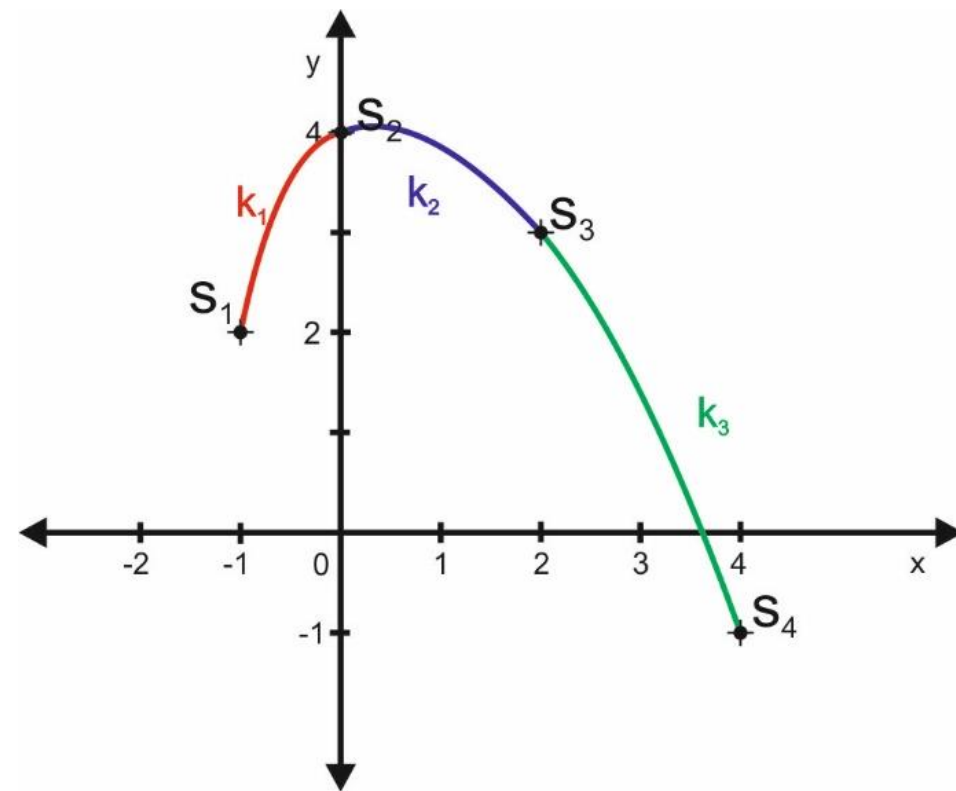
$$x_0 \in \langle x_2; x_3 \rangle$$

$$k_3(x_0) = a_3 + b_3(x_0 - x_3) + c_3(x_0 - x_3)^2 + d_3(x_0 - x_3)^3$$

$$x_0 \in \langle x_3; x_4 \rangle$$

- je potrebné určiť 12 neznámych koeficientov, ktoré nájdeme dosadením hodnôt:

i	1	2	3	4
x_i	-1	0	2	4
y_i	2	4	3	-1



- Podrobné vysvetlenie výpočtu je uvedené v skriptách (Kaňuk, 2015)

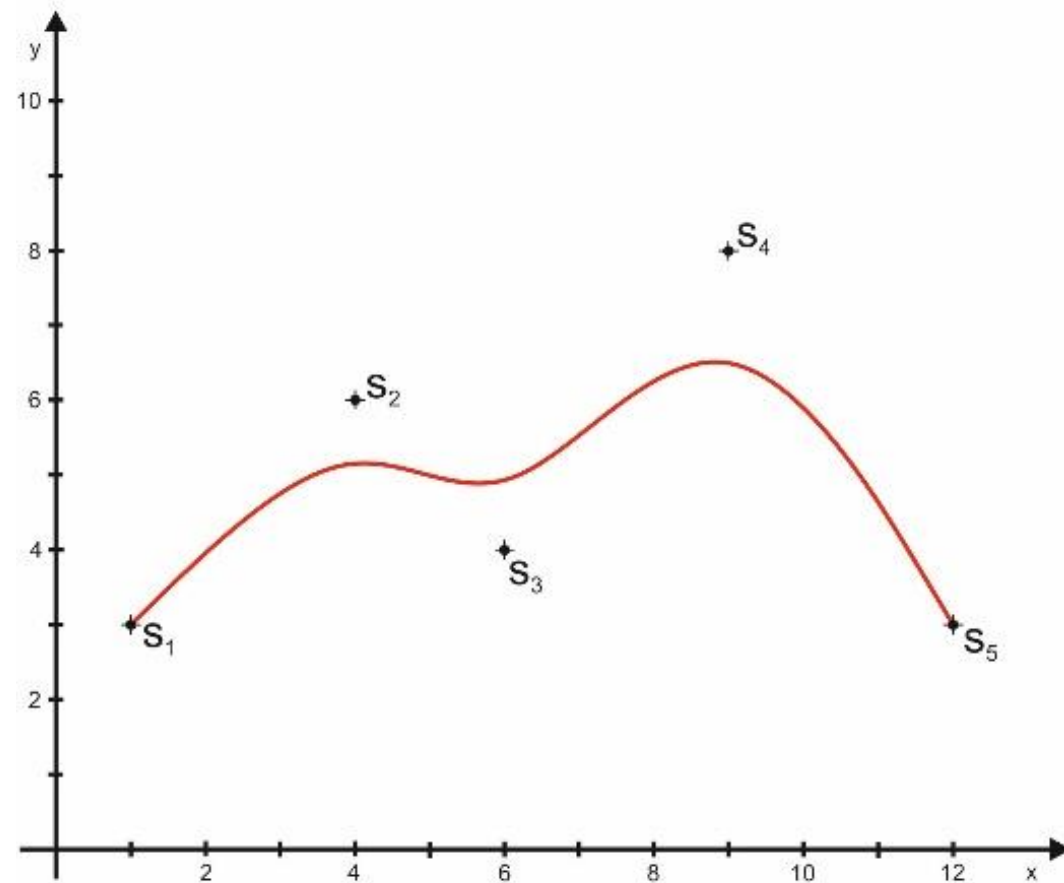
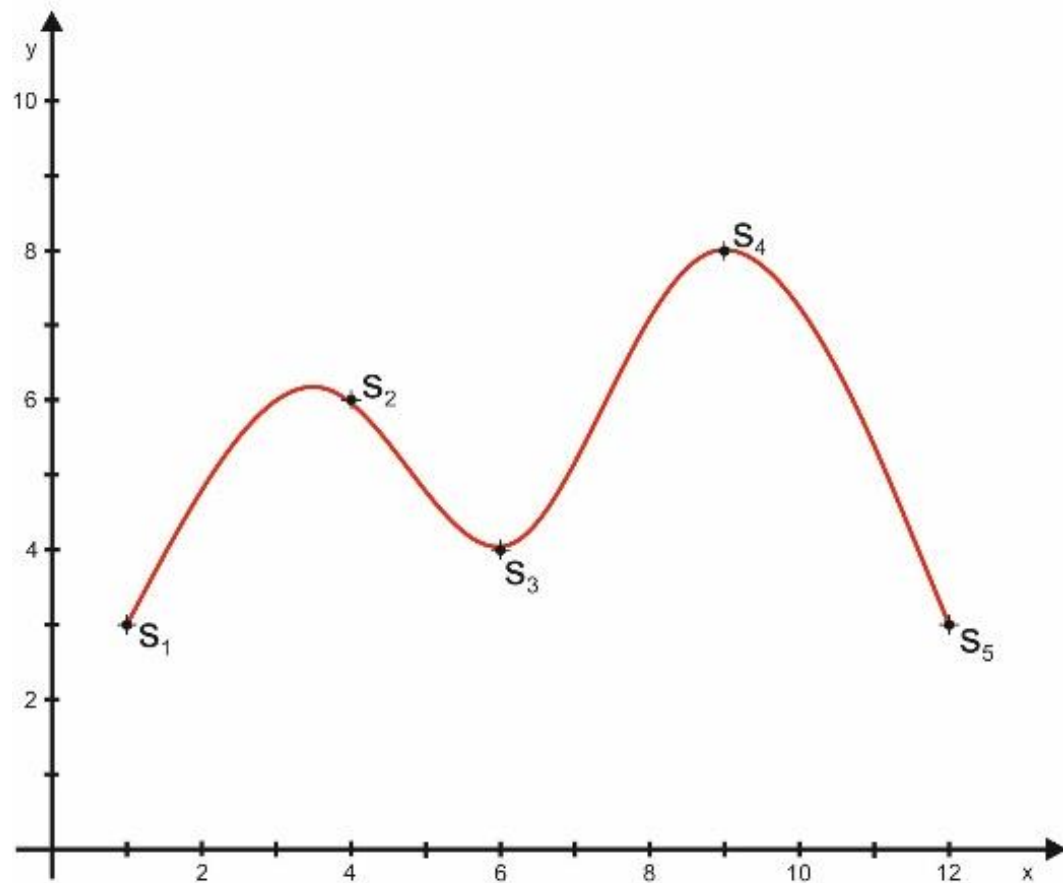
Multivariačný splajn

- veľká časť interpolačných funkcií je založená na vopred známych matematických funkciách, ktoré vychádzajú z určitých znalostí o správaní sa sledovaného geopriestorového javu.
 - napr. IDW vieme vopred, akú funkciu použijeme na odhad hodnôt.
- na odhad hodnôt geopriestorových javov sa však môžeme pozerať aj iným spôsobom, a to tak, že nevieme, aká matematická funkcia ich vyjadruje, avšak vieme akými hodnotami v určitých bodoch funkcia prechádza, a pomocou bodového poľa hodnôt sa pokúšame odhadnúť priebeh funkcie.

Multivariačný splajn

- Splajn patrí medzi interpolačné metódy, ktoré sa snažia nájsť takú funkciu, aby vhodne reprezentovala sledovaný jav.
 - vychádzajú z predpokladu, že modelovaný jav je priestorovo autokorelovaný.
 - Dôležité tiež je nájsť takú funkciu, pre ktorú vieme určiť derivácie vo vybraných bodoch, čo nám napríklad v prípade georeliéfu umožňuje výpočet morfometrických parametrov na základe jej parciálnych derivácií.
 - Splajny predstavujú veľmi robustnú metódu založenú na odhade hodnôt pomocou polynomických funkcií, ktorá si však vyžaduje hlbšie znalosti o vplyve jej parametrov na výsledok
 - Zjednodušene by sme mohli povedať, že splajnové funkcie simulujú správanie sa tenkej elastickej platne, ktorá sa snaží prechádzať bodmi vstupného bodového poľa.
 - Každá časť povrchu je vyjadrená samostatnou polynomickou funkciou.
 - Táto polynomická funkcia je odvodená z lokálnych hodnôt vstupného bodového poľa, pričom jej priebeh je ovplyvnený viacerými podmienkami.
 - Základnou podmienkou je, aby bola zabezpečená spojitosť susedných polynomických funkcií v bodoch vstupného bodového poľa.
 - Týchto podmienok môže byť viacero v závislosti od triedy polynomickej funkcie (napr. v prípade kubickej polynomickej funkcie je definovaná podmienka zhody prvej a druhej derivácie).
 - Výsledkom je hladký povrch.

Multivariačný splajn



Multivariačný splajn

- Základný princíp splajnu pre modelovanie povrchov (spojitých javov) z priestorovo distribuovaných dát je založený na fyzikálnom modeli
 - tenkej elastickej membrány, ktorú sa snažíme preložiť cez body vstupného bodového poľa pomocou definovania podmienok pre ohybnosť funkcie, tzv. tenziu (tension parameter)
 - hladkosť výsledného povrchu (smooth parameter).
- Splajnové metódy sú založené na predpoklade, že interpolačná funkcia:
 - a) prechádza najbližšie ako je len možné bodmi vstupného bodové poľa
 - súčet odchýlok od bodov vstupného bodového poľa je minimálny
 - b) funkcia je najviac hladká, ako len je to možné
 - pre hladkosť funkcie platí nasledovné

$$\sum_{j=1}^N |z^j - S(x^j)|^2 w_j + w_0 I(S) = \text{minimum}$$

Daných je N počet sledovaných javov $z^j, j = 1, \dots, N$ meraných v konkrétnych bodoch $x^{(j)} = (x_1^{(j)}, x_2^{(j)}, \dots, x_d^{(j)})$, $j = 1, \dots, N$ v rámci istého regiónu d – dimenzionálneho priestoru a hľadáme funkciu $S(x)$

w_j a w_0 sú pozitívne váhové faktory

$I(S)$ je seminorma hladkosti (smooth seminorm)

Multivariačný splajn

- Riešenie tohto problému by sme mohli vyjadriť ako sumu dvoch komponentov

$$S(x) = T(x) + \sum_{i=1}^n \lambda_i R(x, x^{(j)})$$

$T(x)$ je 'trendová' funkcia

$R(x, x^{(j)})$ je základná radiálna funkcia v explicitnej forme, ktorá je závislá na voľbe $I(S)$

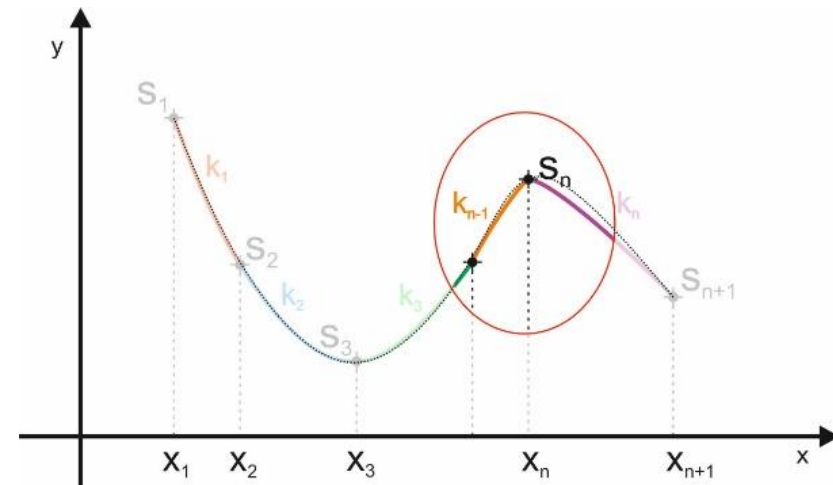
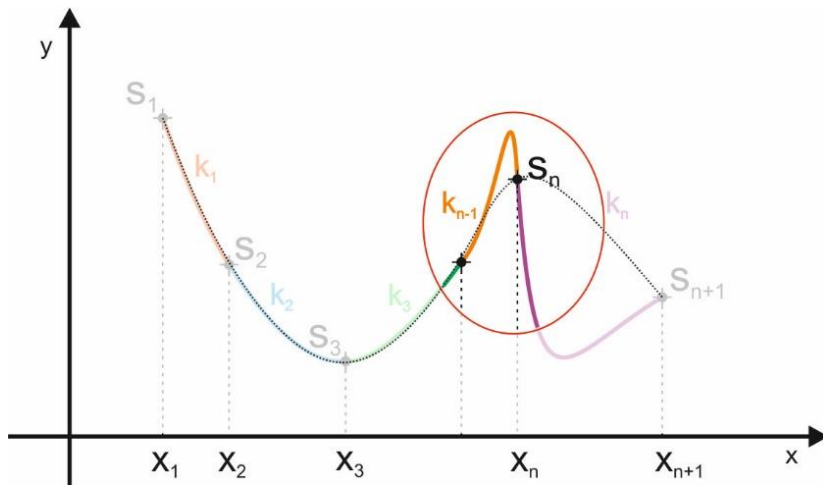
- ďalšie podrobnosti sú uvedené v skriptách (Kaňuk, 2015)

Multivariačný splajn - parametre

- parameter tenzie φ kontroluje vzdialenosť, cez ktorú daný bod vplýva na výsledný povrch. Optimálne nastavenie tohto parametra by sme si mohli predstaviť ako ladenie charakteru výsledného povrchu medzi membránou a tenkou doskou.
- ak predpokladáme, že parameter zhladenia (smooth parameter) bude rovnaký pre všetky body, potom platí

φ – všeobecný výraz pre parameter tenzie (φ - čítaj phi)

- parameter zhladenia (smooth parameter)

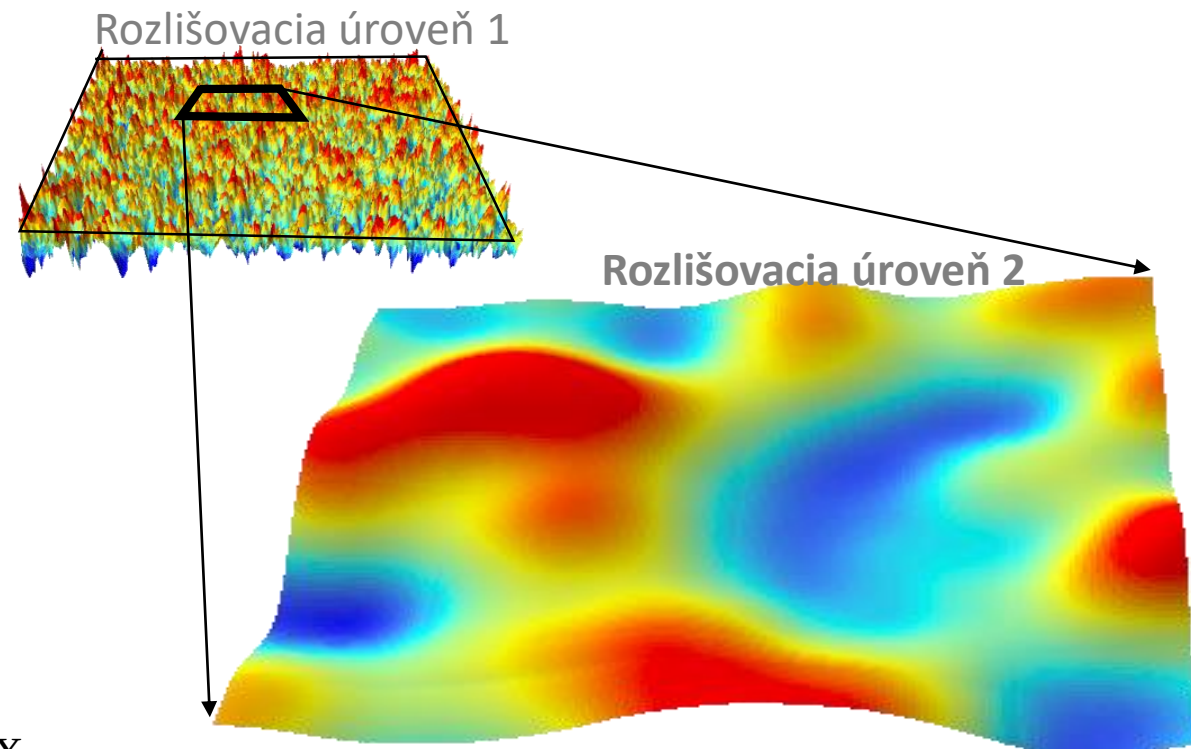
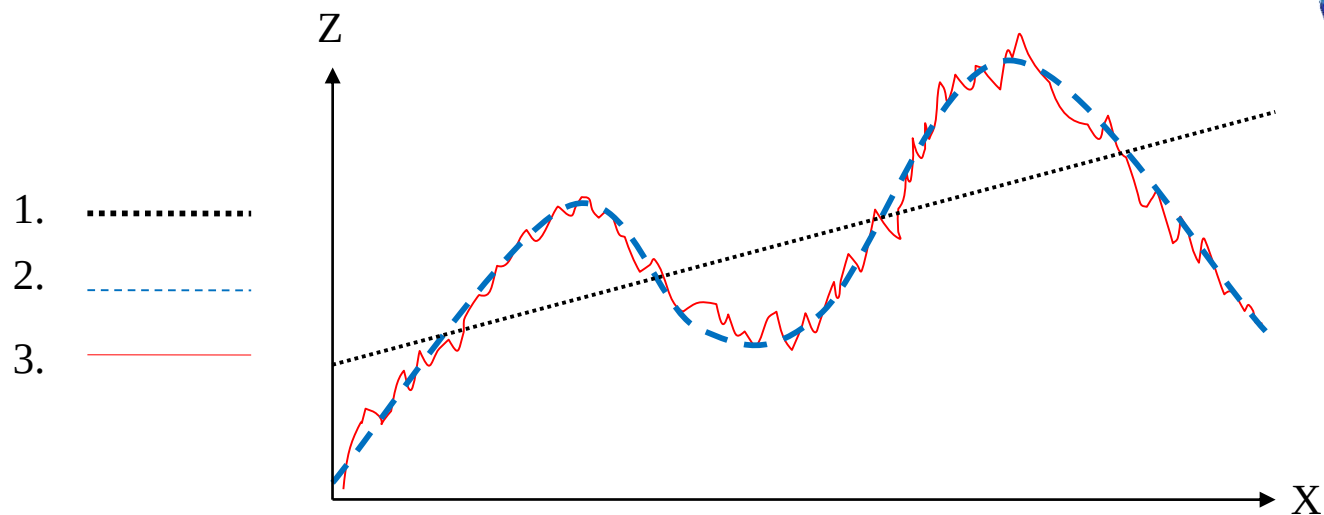


Multivariačný splajn

Hodnoty priestorovej premennej (javu) $Z(x)$ možno rozložiť na:
deterministickú zložku $m(x)$ a náhodnú (geoštatistickú) zložku $R(x)$

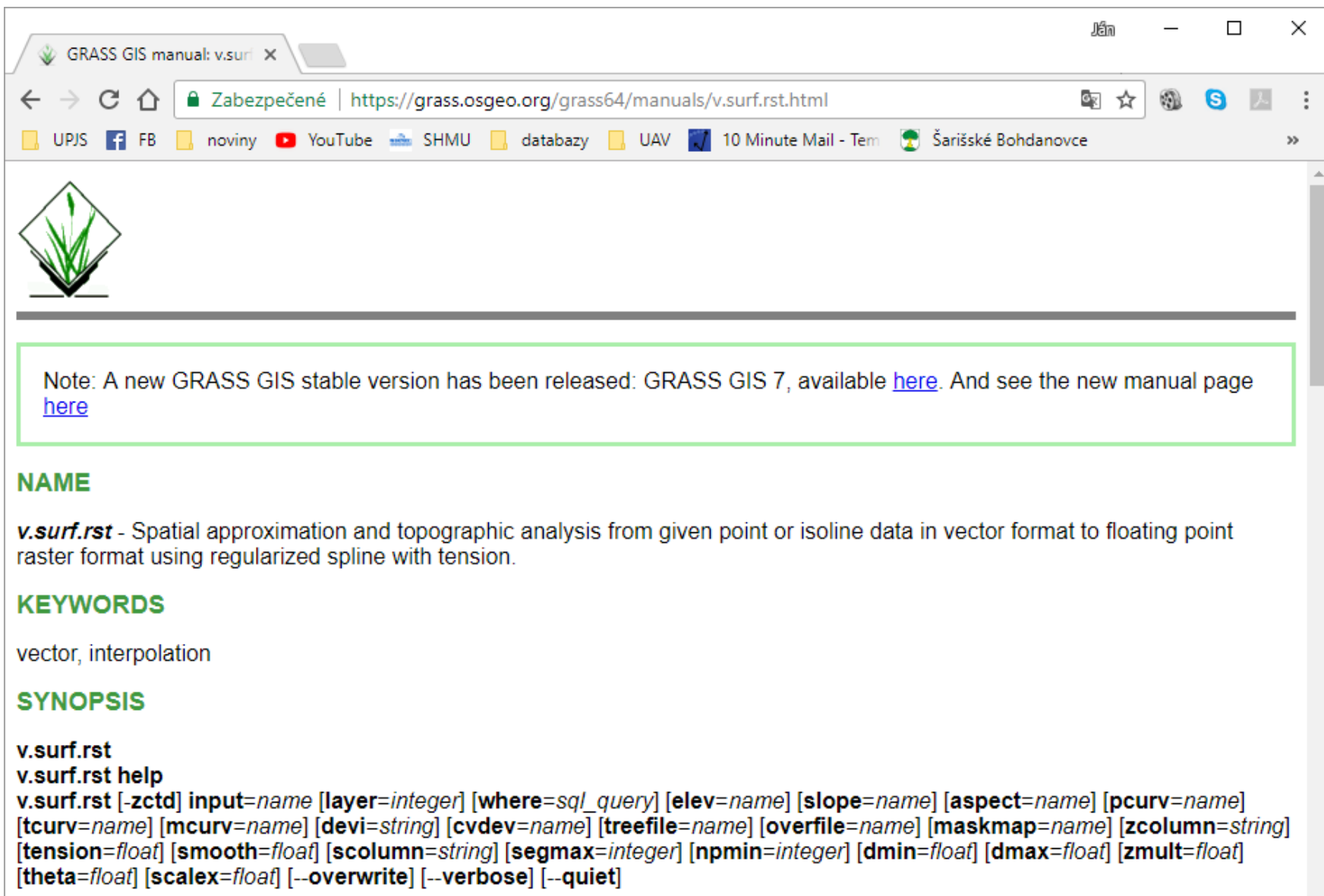
$$Z(x) = m(x) + R(x)$$

$dáta =$ systematická + priestorovo + náhodná
zložka autokorelovaná zložka zložka



v.surf.rst

Modul implementovaný do softvéru GRASS GIS



The screenshot shows a web browser window with the title "GRASS GIS manual: v.surf.rst". The address bar shows the URL "https://grass.osgeo.org/grass64/manuals/v.surf.rst.html". The browser's toolbar includes various icons and a search bar. The main content area features the GRASS GIS logo, a note about the release of GRASS GIS 7, and sections for the module's name, keywords, synopsis, and command-line options.

Note: A new GRASS GIS stable version has been released: GRASS GIS 7, available [here](#). And see the new manual page [here](#)

NAME

v.surf.rst - Spatial approximation and topographic analysis from given point or isoline data in vector format to floating point raster format using regularized spline with tension.

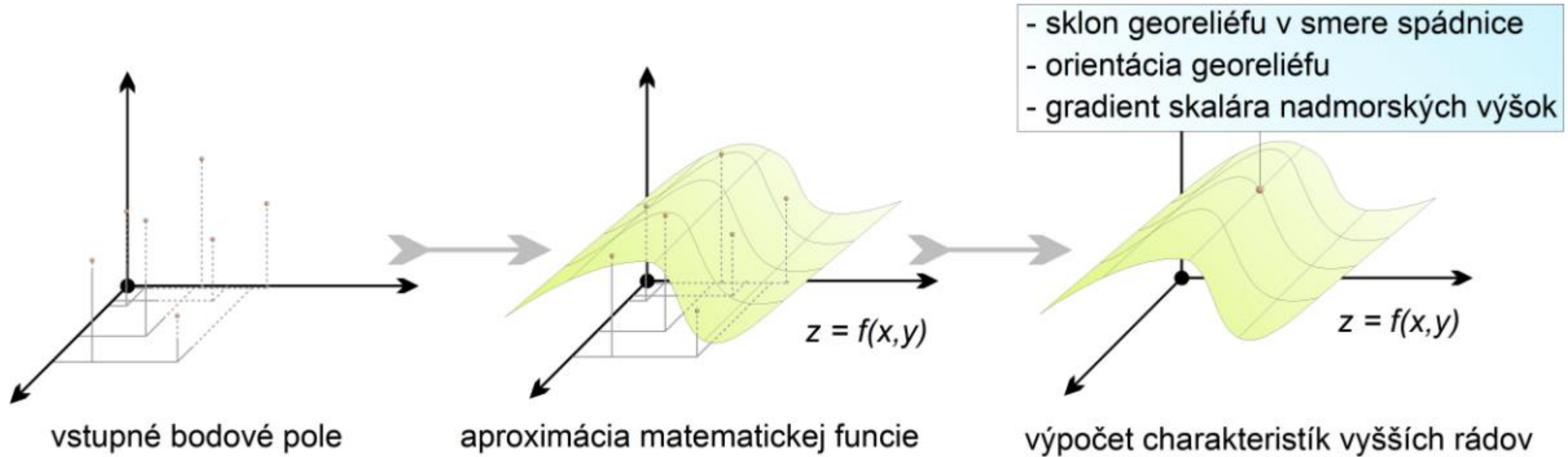
KEYWORDS

vector, interpolation

SYNOPSIS

v.surf.rst
v.surf.rst help
v.surf.rst [-zctd] input=name [layer=integer] [where=sql_query] [elev=name] [slope=name] [aspect=name] [pcurv=name] [tcurv=name] [mcurv=name] [devi=string] [cvdev=name] [treefile=name] [overfile=name] [maskmap=name] [zcolumn=string] [tension=float] [smooth=float] [scolumn=string] [segmax=integer] [npmin=integer] [dmin=float] [dmax=float] [zmult=float] [theta=float] [scalex=float] [--overwrite] [--verbose] [--quiet]

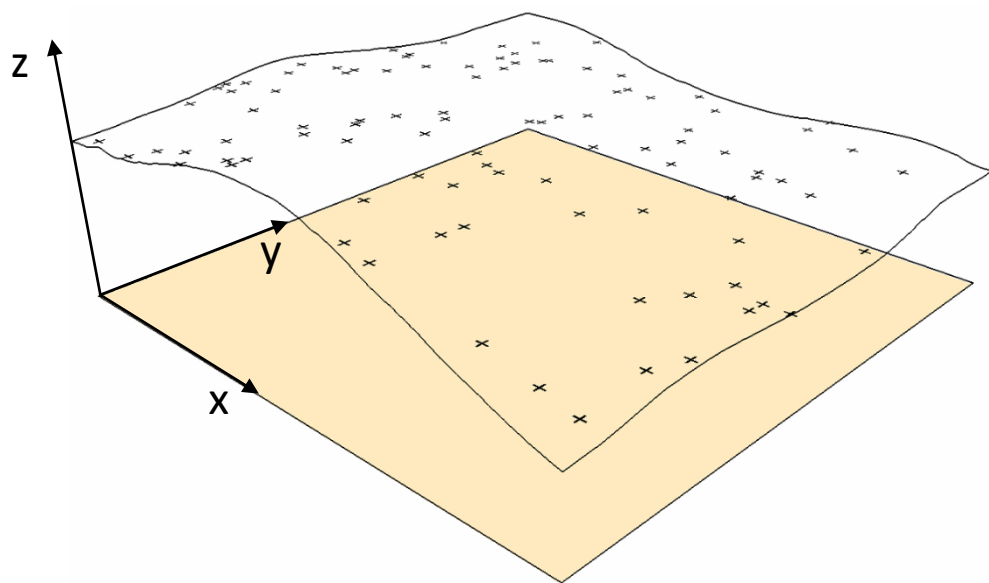
Princíp výpočtu



v.surf.rst

Vstup

= vstupné bodové pole nadmorských výšok



Parameters:

input=*name*
Name of input vector map

layer=*integer*
Layer number
If set to 0, z coordinates are used. (3D vector only)
Default: 1

where=*sql_query*
WHERE conditions of SQL statement without 'where' keyword
Example: income < 1000 and inhab >= 10000

elev=*name*
Output surface raster map (elevation)

slope=*name*
Output slope raster map

aspect=*name*
Output aspect raster map

pcurv=*name*
Output profile curvature raster map

tcurv=*name*
Output tangential curvature raster map

mcurv=*name*
Output mean curvature raster map

devi=*string*
Output deviations vector point file

cvdev=*name*
Output cross-validation errors vector point file

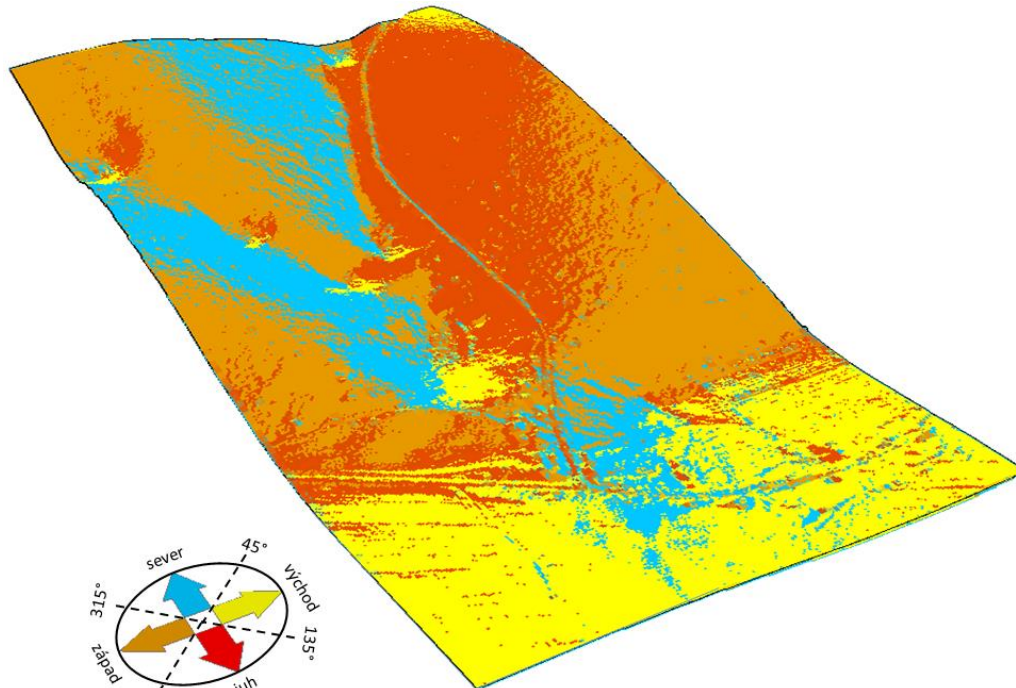
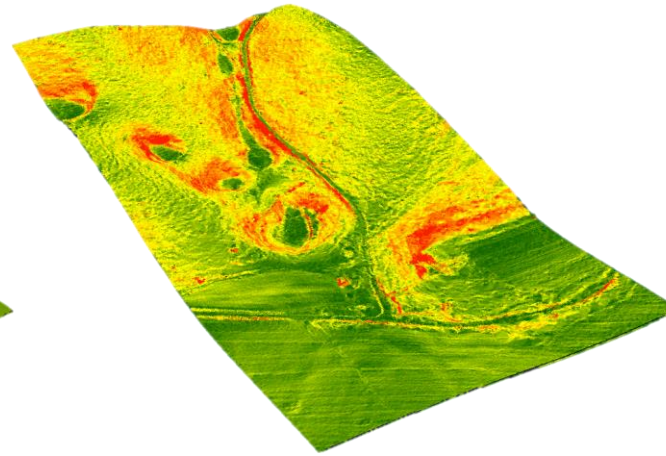
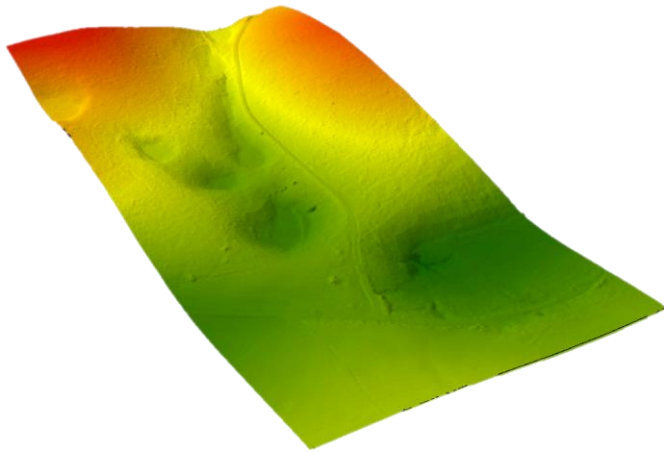
treefile=*name*
Output vector map showing quadtree segmentation

overfile=*name*
Output vector map showing overlapping windows

maskmap=*name*
Name of the raster map used as mask

v.surf.rst

výstup



Parameters:

input=*name*

Name of input vector map

layer=*integer*

Layer number

If set to 0, z coordinates are used. (3D vector only)

Default: 1

where=*sql_query*

WHERE conditions of SQL statement without 'where' keyword

Example: income < 1000 and inhab >= 10000

elev=*name*

Output surface raster map (elevation)

slope=*name*

Output slope raster map

aspect=*name*

Output aspect raster map

pcurv=*name*

Output profile curvature raster map

tcurv=*name*

Output tangential curvature raster map

mcurv=*name*

Output mean curvature raster map

devi=*string*

Output deviations vector point file

cvdev=*name*

Output cross-validation errors vector point file

treefile=*name*

Output vector map showing quadtree segmentation

overfile=*name*

Output vector map showing overlapping windows

maskmap=*name*

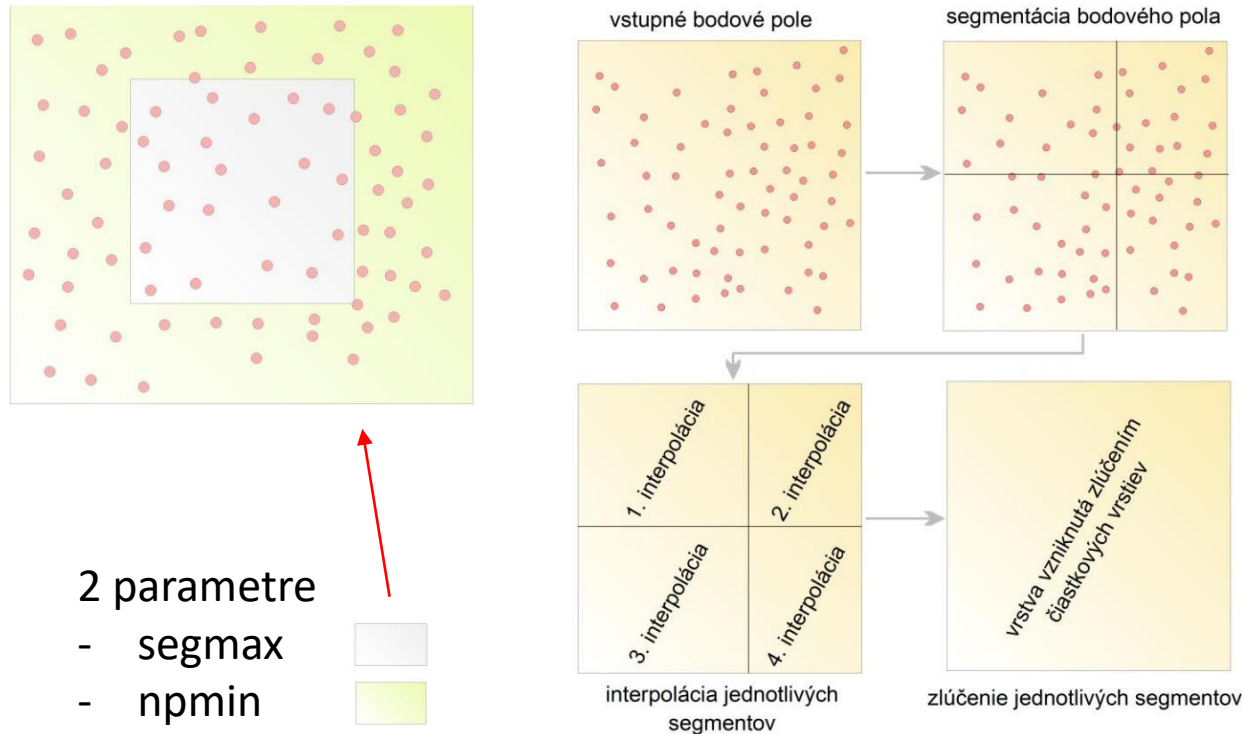
Name of the raster map used as mask

v.surf.rst

- dávkovo pracuje so 700 vstupnými bodmi naraz
- v prípade väčšieho počtu vstupného bodového poľa dochádza ku segmentácii

Problém súvisiaci so **segmentáciou**

- zabezpečenie hladkého spojenia jednotlivých segmentov



Počet bodov v segmente (segmax) je nižší ako počet bodov pre širšiu oblasť (npmin)

Parameters:

mcurv=name
Output mean curvature raster map

devi=string
Output deviations vector point file

cvdev=name
Output cross-validation errors vector point file

treefile=name
Output vector map showing quadtree segmentation

overfile=name
Output vector map showing overlapping windows

maskmap=name
Name of the raster map used as mask

zcolumn=string
Name of the attribute column with values to be used for approximation (if layer>0)

tension=float
Tension parameter
Default: 40.

smooth=float
Smoothing parameter

scolumn=string
Name of the attribute column with smoothing parameters

segmax=integer
Maximum number of points in a segment
Default: 40

npmin=integer
Minimum number of points for approximation in a segment (>segmax)
Default: 300

dmin=float
Minimum distance between points (to remove almost identical points)
Default: 0.500000

dmax=float
Maximum distance between points on isoline (to insert additional points)
Default: 2.500000

zmult=float
Conversion factor for values used for approximation
Default: 1.0

theta=float
Anisotropy angle (in degrees counterclockwise from East)

scalex=float
Anisotropy scaling factor

v.surf.rst

- Pri kombinácii viacerých parametrov je možné analyzovať a filtrovať vstupné bodové pole, prevzorkovať ho a použiť pre odvodenie povrchov s požadovanými vlastnosťami

Parameters:

mcurv=*name*
Output mean curvature raster map

devi=*string*
Output deviations vector point file

cvdev=*name*
Output cross-validation errors vector point file

treefile=*name*
Output vector map showing quadtree segmentation

overfile=*name*
Output vector map showing overlapping windows

maskmap=*name*
Name of the raster map used as mask

zcolumn=*string*
Name of the attribute column with values to be used for approximation (if layer>0)

tension=*float*
Tension parameter
Default: 40.

smooth=*float*
Smoothing parameter

scolumn=*string*
Name of the attribute column with smoothing parameters

segmax=*integer*
Maximum number of points in a segment
Default: 40

npmin=*integer*
Minimum number of points for approximation in a segment (>segmax)
Default: 300

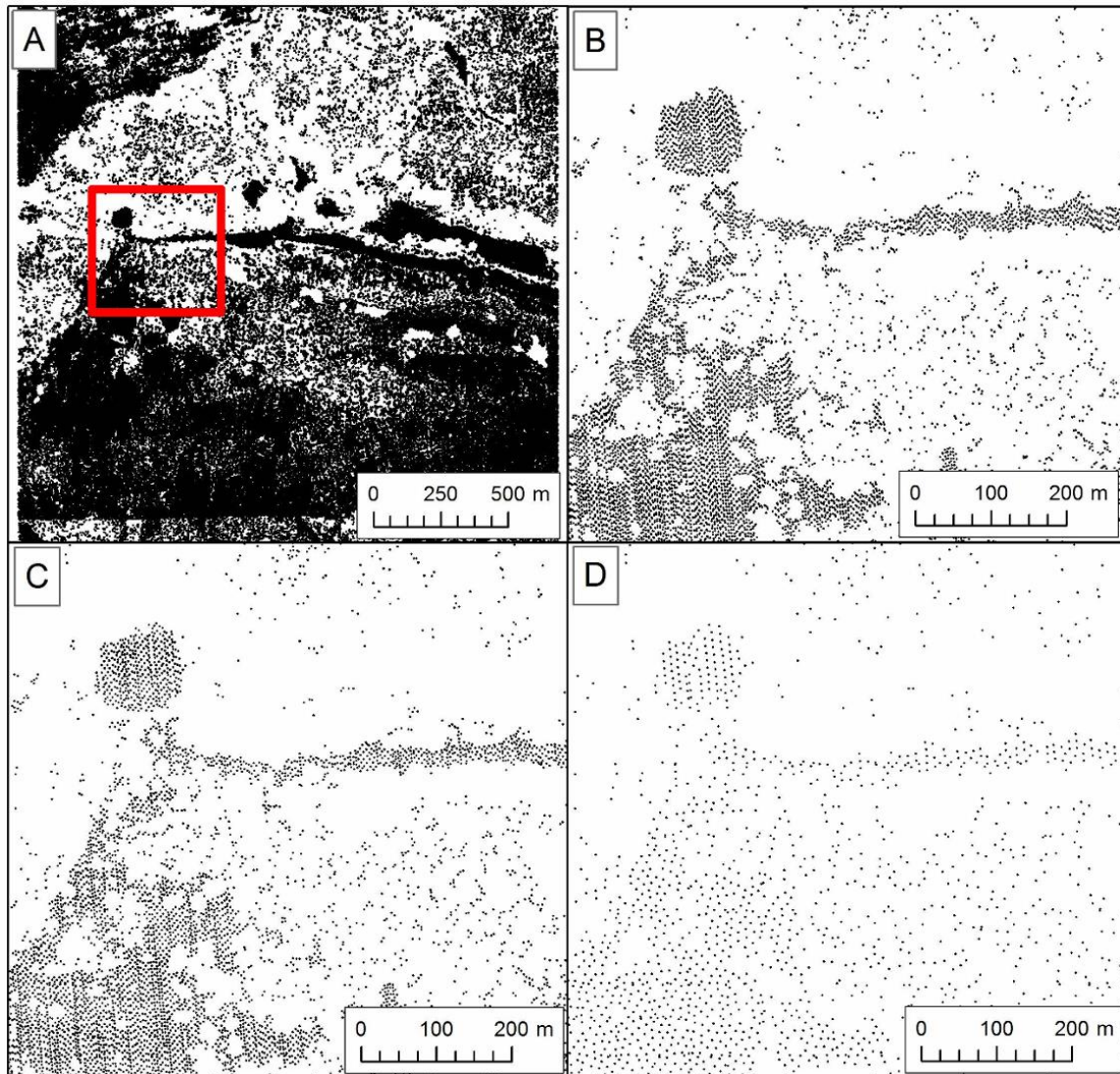
dmin=*float*
Minimum distance between points (to remove almost identical points)
Default: 0.500000

dmax=*float*
Maximum distance between points on isoline (to insert additional points)
Default: 2.500000

zmult=*float*
Conversion factor for values used for approximation
Default: 1.0

theta=*float*
Anisotropy angle (in degrees counterclockwise from East)

scalex=*float*
Anisotropy scaling factor



Dáta z LLS použité pre tvorbu DMR klasifikované ako (2) terén:

A – záujmové územie

B – detail na územie v červenom rámciku

C – redukované mračno bodov s minimálnimi rozstupmi bodov 2 m

D – redukované mračno bodov s minimálnimi rozstupmi bodov 6 m

Nehomogénna distribúcia dát spôsobuje problémy pri interpolácii.

Jedným z riešení je selektívna filtrácia vstupných dát.

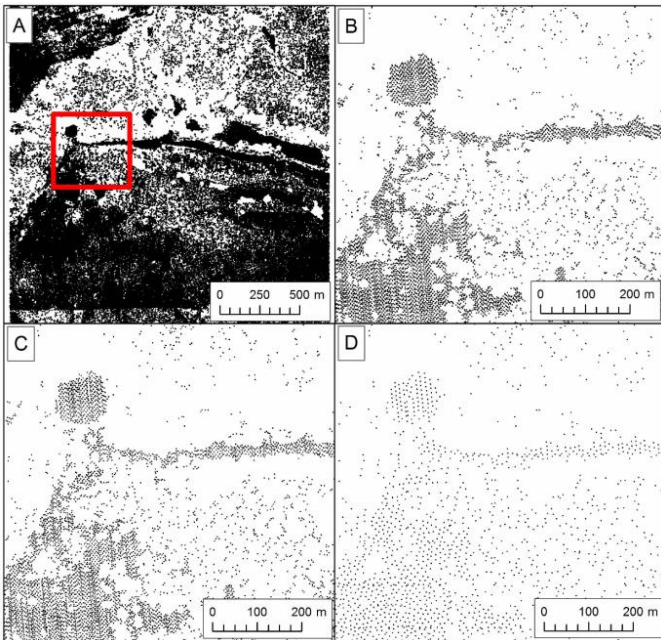
v.surf.rst

Selektívna filtrácia bodov identifikujúca lokálne maximá a minimá (devi)

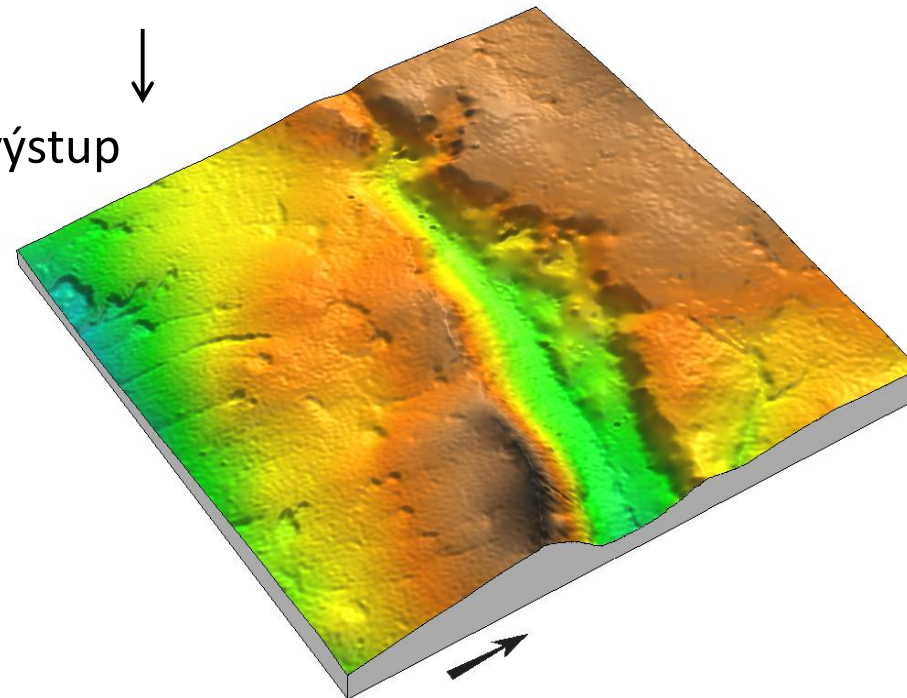
Interpolácia pomocou regularizovaného splajnu s tenziou (v.surf.rst) v GRASS GIS

- Optimalizácia segmentácie (segmax a npmin)
- Nastavenie minimálnej vzdialenosti medzi bodmi (dmin)

vstupné data



výstup



$$f(\mathbf{x}) = a_1 + \sum_{j=1}^N \lambda_j \{ -[E_1(\rho) + \ln \rho + C_E] \}$$



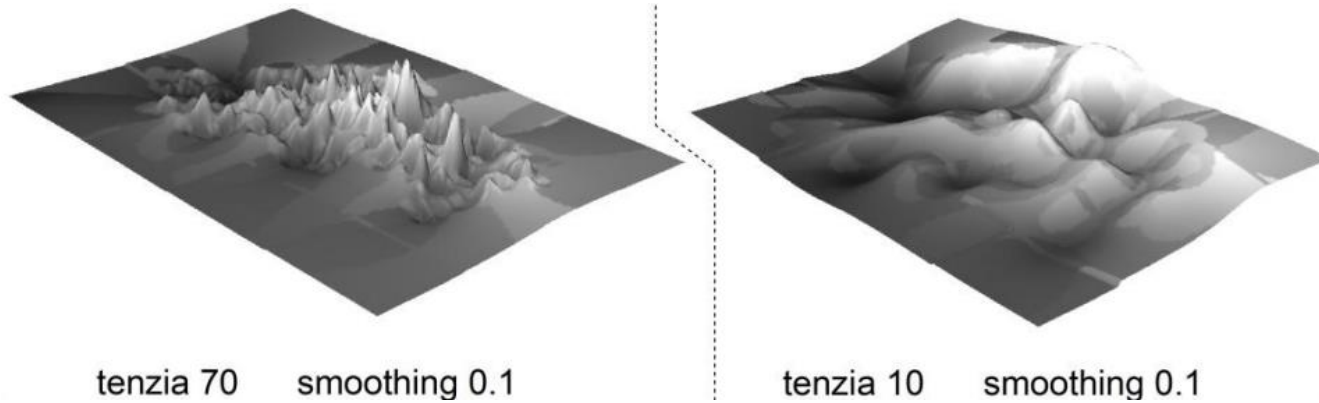
HOFIERKA, J., GALLAY, M., KAŇUK, J., (2013): Spatial Interpolation of Airborne Laser Scanning Data with Variable Data Density. *Proceedings of the 26th International Cartographic Conference*, August 25-30, 2013, Dresden, Germany, 1-12.

v.surf.rst

- Tension – ide o pružnosť (ohybnosť, tuhosť) aproximačnej funkcie

Čím je vyššia pružnosť, tým má funkcia väčšiu voľnosť a možnosť adaptovať sa na vstupné bodové pole

Čím má nižšiu pružnosť (vyššiu tuhosť), tým je funkcia viac hladká a nemá možnosť prechádzať zadanými bodmi



- Smooth - zhladenie

Parameters:

mcurv=*name*
Output mean curvature raster map

devi=*string*
Output deviations vector point file

cvdev=*name*
Output cross-validation errors vector point file

treefile=*name*
Output vector map showing quadtree segmentation

overfile=*name*
Output vector map showing overlapping windows

maskmap=*name*
Name of the raster map used as mask

zcolumn=*string*
Name of the attribute column with values to be used for approximation (if layer>0)

tension=*float*
Tension parameter
Default: 40.

smooth=*float*
Smoothing parameter

scolumn=*string*
Name of the attribute column with smoothing parameters

segmax=*integer*
Maximum number of points in a segment
Default: 40

npmin=*integer*
Minimum number of points for approximation in a segment (>segmax)
Default: 300

dmin=*float*
Minimum distance between points (to remove almost identical points)
Default: 0.500000

dmax=*float*
Maximum distance between points on isoline (to insert additional points)
Default: 2.500000

zmult=*float*
Conversion factor for values used for approximation
Default: 1.0

theta=*float*
Anisotropy angle (in degrees counterclockwise from East)

scalex=*float*
Anisotropy scaling factor

Obsah predmetu

1. Úvod do priestorových analýz
 2. Databázy, výbery z databáz a dopyty – PH1
 3. Vzdialenostné analýzy a mapová algebra – PH1
 4. Analýza distribúcie priestorových dát a priestorová autokorelácia – PH2
- Priebežné hodnotenie 1
5. Jadrové analýzy hustoty, Geograficky vážená regresia - PH2
 6. Vybrané metódy interpolácie - Lineárna interpolácia, Thiessenove polygóny, Metóda najbližšieho suseda – PH3

Priebežné hodnotenie 2,

7. Vybrané metódy interpolácie – Metóda prirodzeného suseda, bilineárna interpolácia, metóda inverzne váženej vzdialenosti – PH3
8. Vybrané metódy interpolácie - Metóda trendového povrchu, multivariačný splajn

Priebežné hodnotenie 3,

9. Vybrané metódy interpolácie – Triangulácia - PH4
10. Vybrané metódy interpolácie - Kriging
11. Hodnotenie kvality interpolácie, Morfometria a vzťah k aplikáciám DMR – PH4
12. Modelovanie slnečného žiarenia

Priebežné hodnotenie 4,

13. Modelovanie toku vody a erózie (USPED, SIMWE)
14. 3D interpolácia – východiská