

Metamodeling

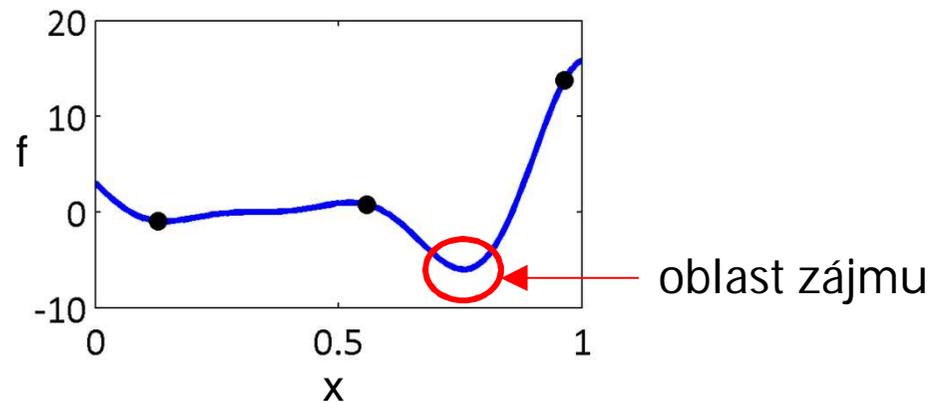
- Nejmodernější oblast optimalizace
- Určena zejména pro praktické aplikace s velkými výpočetními nároky
- Vychází z myšlenky, že reálné optimalizační problémy nejsou sice konvexní, ale jsou do značné míry „hladké“
- Cíl: najít aproximaci (meta-model) M problému (modelu) P tak aby:
 - M bylo méně výpočetně náročné než P
 - Aby minimum M a P bylo totožné

Meta-modely v optimalizaci

- Náhrada účelové funkce
 - Minimum v meta-modelu je rovné minimu původního modelu

Příklad optimalizační úlohy:

$$\min f(x) = (6x - 2)^2 \sin(12x - 4)$$



Meta-modely v optimalizaci

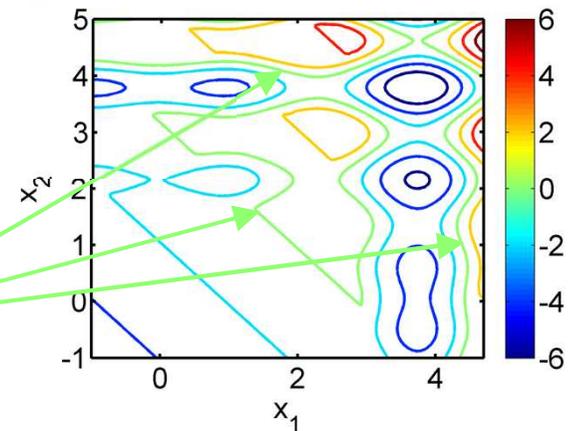
- Náhrada omezující funkce
 - Hyperplocha rozdělující doménu na oblast přípustnou a nepřípustnou je co nejvíce shodná jak v původním modelu, tak v meta-modelu

Příklad optimalizační úlohy:

$$\min f(x) = (x_1 - 3.7)^2 + (x_2 - 4)^2$$

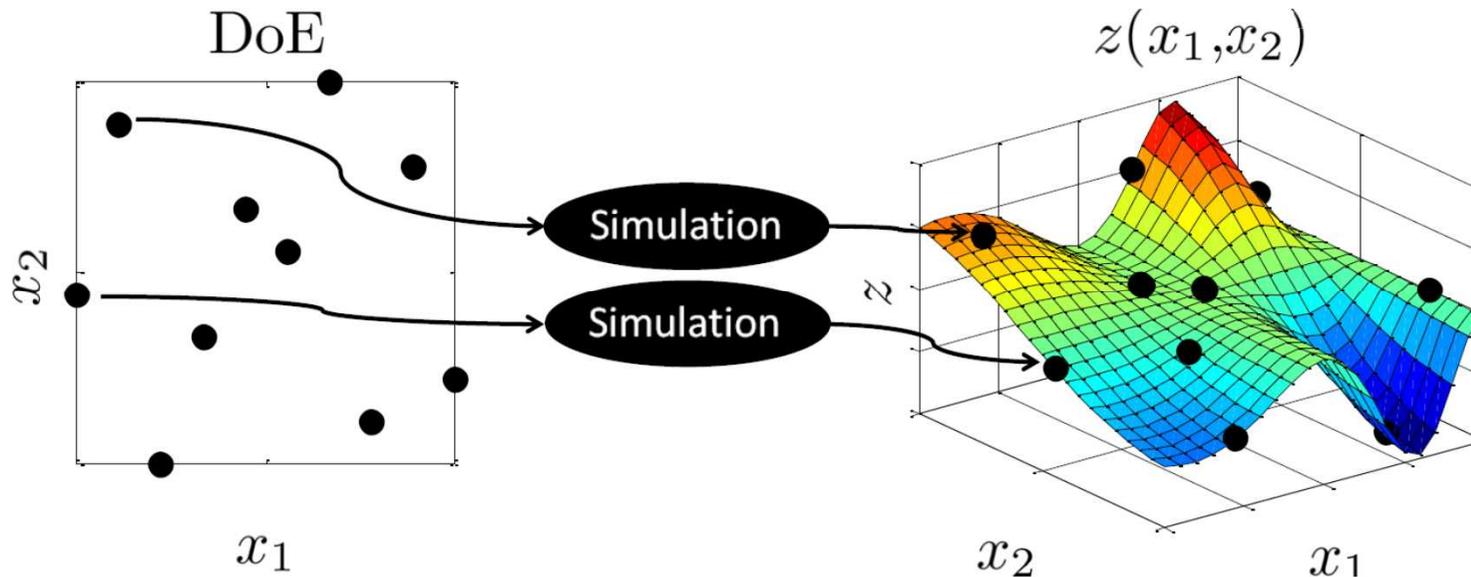
$$\text{vzhledem k } \begin{cases} -x_1 \sin(4x_1) - 1.1x_2 \sin(2x_2) \geq 0 \\ x_1 + x_2 - 3 \geq 0 \end{cases}$$

Oblast zájmu (křivka pro 2D)



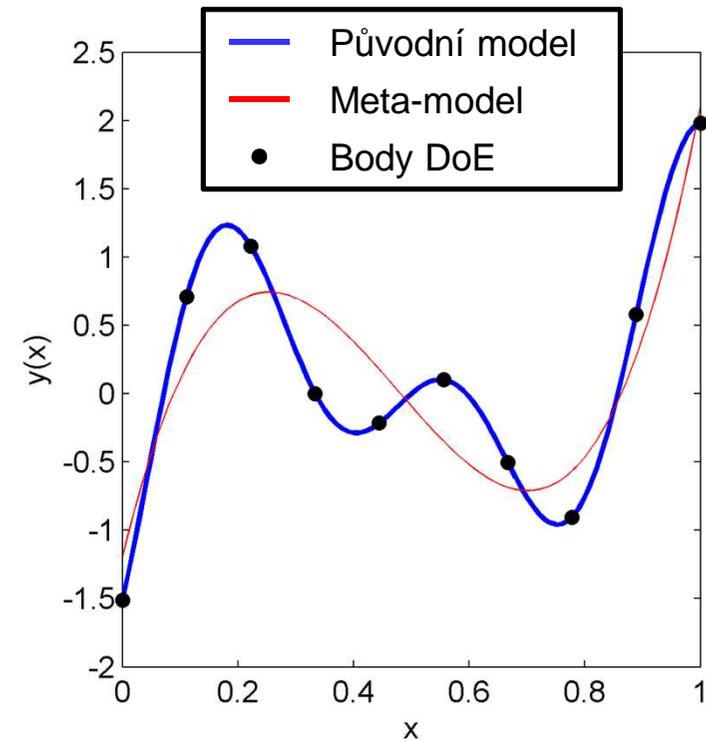
Tvorba meta-modelu

- Původní model je potřeba vyhodnotit v některých bodech, tzv. bodech návrhu experimentu (DoE z „*Design of experiment*“)
- Hodnoty odezvy původního modelu se proloží křivkou



Aproximační meta-modely

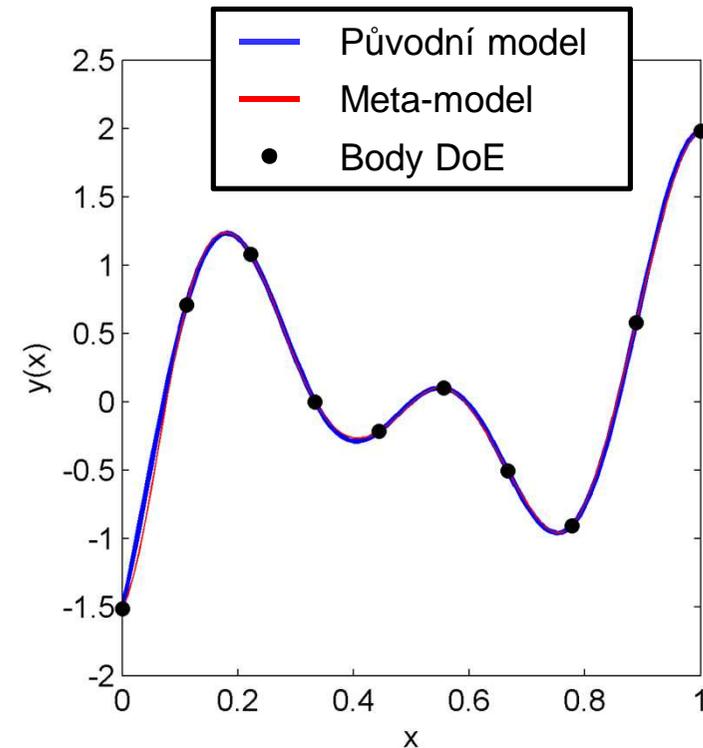
- Též zvané regresní modely
- Odezva získaná pomocí meta-modelu a původního modelu se liší
- Zástupci:
 - Lineární regrese
 - Kvadratická, polynomiální regrese
 - Moving Least Squares
 - Polynomiální chaos
 - atd.



Původní model: $y(x) = (4x - 2)\sin(12x - 4)$
Typ meta-modelu: Polynomiální chaos
s Legendrovými polynomy 3. stupně

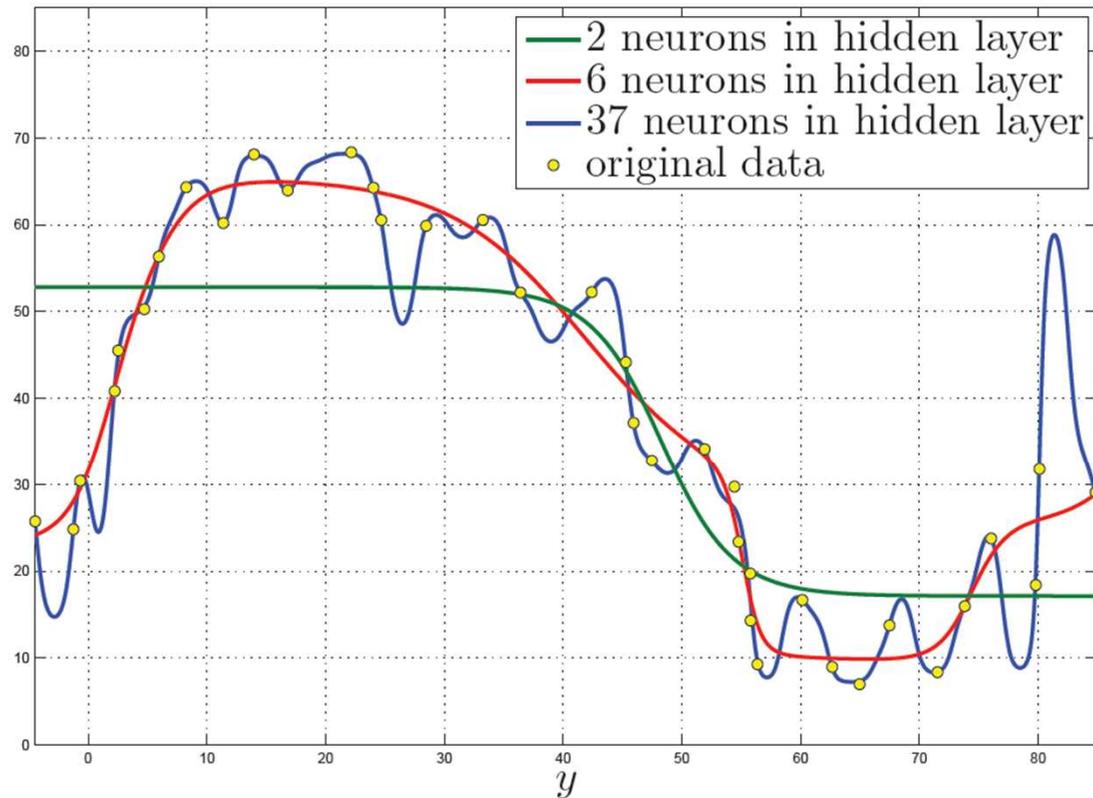
Interpolační meta-modely

- Odezva získaná pomocí meta-modelu a původního modelu je totožná
- Zástupci:
 - Radiální báze
 - Kriging
 - Support vector Machines



Původní model: $y(x) = (4x - 2)\sin(12x - 4)$
Typ meta-modelu: Radiální báze

Problém podučení a přeučení



Approximation of data by multi-layer perceptron with different topology.

Problém podučení a přeučení

- Řešení: tři nezávislé sady dat
 - Trénovací
 - Testovací
 - Validační

Struktura obecného meta-modelu

DoE	Výběr modelu	Nastavení modelu	Příklad techniky
Faktoriální	polynom	Regrese minima čtverců	Metoda plochy odezvy
Centrální kompozitní	Spliny	Váhová regrese minima čtverců	
D-optimální	Realizace náhodného pole	Nejlepší lineární prediktor	Kriging
Plně náhodné	Množina funkcí a terminálů	Genetický algoritmus	Genetické programování
Latin Hypercube	Síť neuronů	Zpětná propagace	BP Neuronové sítě
Ručně vybrané	Rozhodovací strom		
Ortogonální pole	Funkce s radiální bází	Minimalizace Entropie	Induktivní učení

Návrh experimentů

Metody generování:



*Náhodný
návrh*

*LHS
návrh*

*Optimální
návrh*

*Optimální
LHS návrh*

*Faktoriální
návrh*

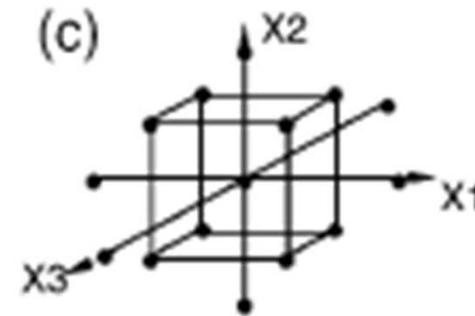
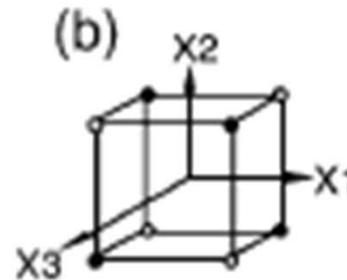
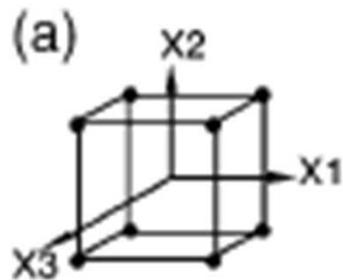
**Řídká
mřížka**
KPU

**Řídká
mřížka**
GQU

DoE – design of experiments

(Návrh experimentů)

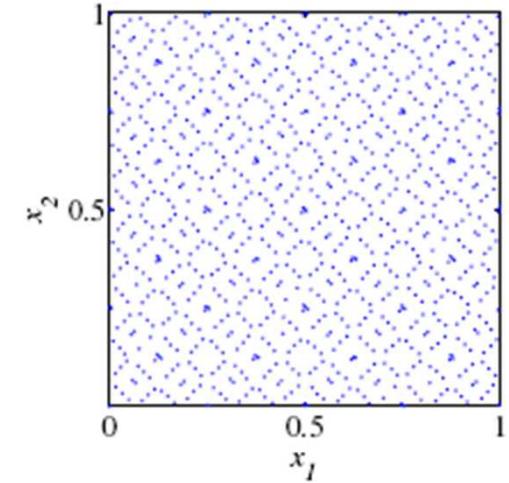
- Faktoriální návrhy



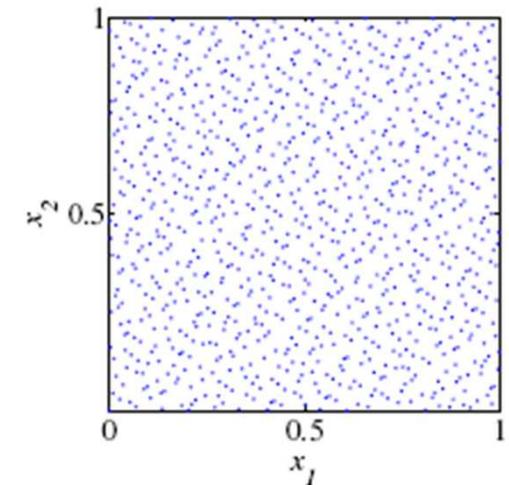
- Plně faktoriální
- Částečně faktoriální
- Kompozitní

DoE

- Náhodná čísla
 - pseudo-náhodná
 - matematické posloupnosti
 - např. funkce `rand()`
 - kvazi-náhodná
 - náhodná, ale rovnoměrně rozložená
 - matematické posloupnosti (Sobolovy sekvence)
 - metody založené na kvazi-verzi metody Monte Carlo (Latin Hypercube Sampling)



(b) Sobol - 2D - 1000 bodů.



(e) Halton - 2D - 1000 bodů.

DoE

- Metoda Monte Carlo
 - simulační metoda využívající pseudo-náhodná čísla
 - generuje náhodné vektory s předepsaným náhodným rozdělením
- Metoda kvazi-Monte Carlo
 - využívá kvazi-náhodná čísla

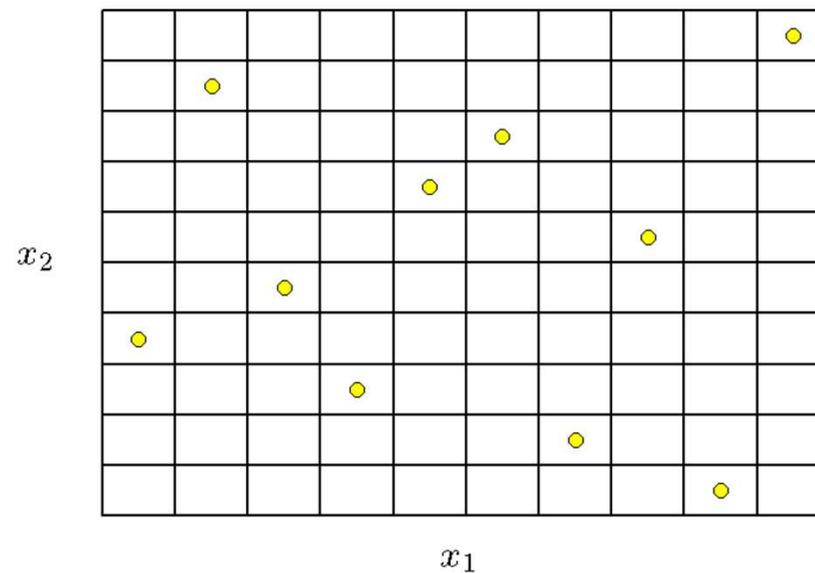
Metoda LHS

- Latin Hypercube Sampling
 - simulační metoda typu (kvazi-)Monte Carlo
 - využívá kvazi-náhodná čísla
 - vyžaduje řádově méně simulací než Monte Carlo metoda
 - rozdělení definičního na N_{sim} stejně pravděpodobných disjunktálních intervalů
 - výběr vzorků ze středů intervalů
 - změna pořadí hodnot vzorků, nikoliv změna hodnot

Metoda LHS

realizace

x_n	3	4	5	6	7	8	9
	-4	-3	-2	-1	0	1	2
...	1	2	3	4	5	6	7
	32	34	36	38	40	42	44
	4	5	6	7	8	9	10
x_2	73	76	79	82	85	88	91
x_1	6	7	8	9	10	11	12



Optimal Latin Hypercube Sampling

- cíl: optimalizovat „rovnoměrnost rozdělení“ navržených vektorů
- metody:
 - maximalizace entropie
 - maximalizace minimální vzdálenosti mezi body
 - kritérium založené na potenciální energii
 - minimalizace rozdílu mezi získanou a předepsanou korelační maticí
- optimalizační algoritmus: simulované žíhání

Rovnoměrné rozprostření návrhu

- Audze – Eglais (AE) [P. Audze, V. Eglais, 1977]

$$E^{\text{AE}} = \sum_{i=1}^n \sum_{j=i+1}^n \frac{1}{L_{ij}^2}, \quad \text{- potenciální energie}$$

- Euklidovská maximin vzdálenost (EMM) [M. Johnson, 1990]

$$E^{\text{EMM}} = - \min\{\dots, L_{ij}, \dots\}, \quad i = 1 \dots n, \quad j = (i + 1) \dots n$$

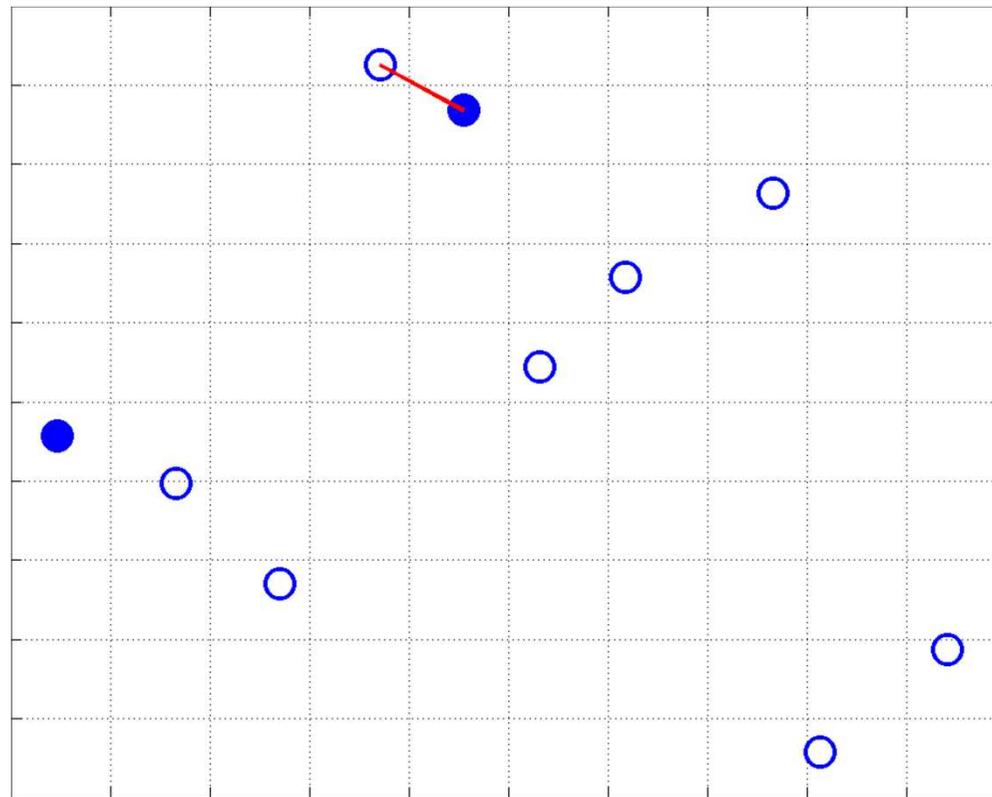
- Modifikovaná L_2 diskrepance (ML2) [T. M. Cioppa, 2007]

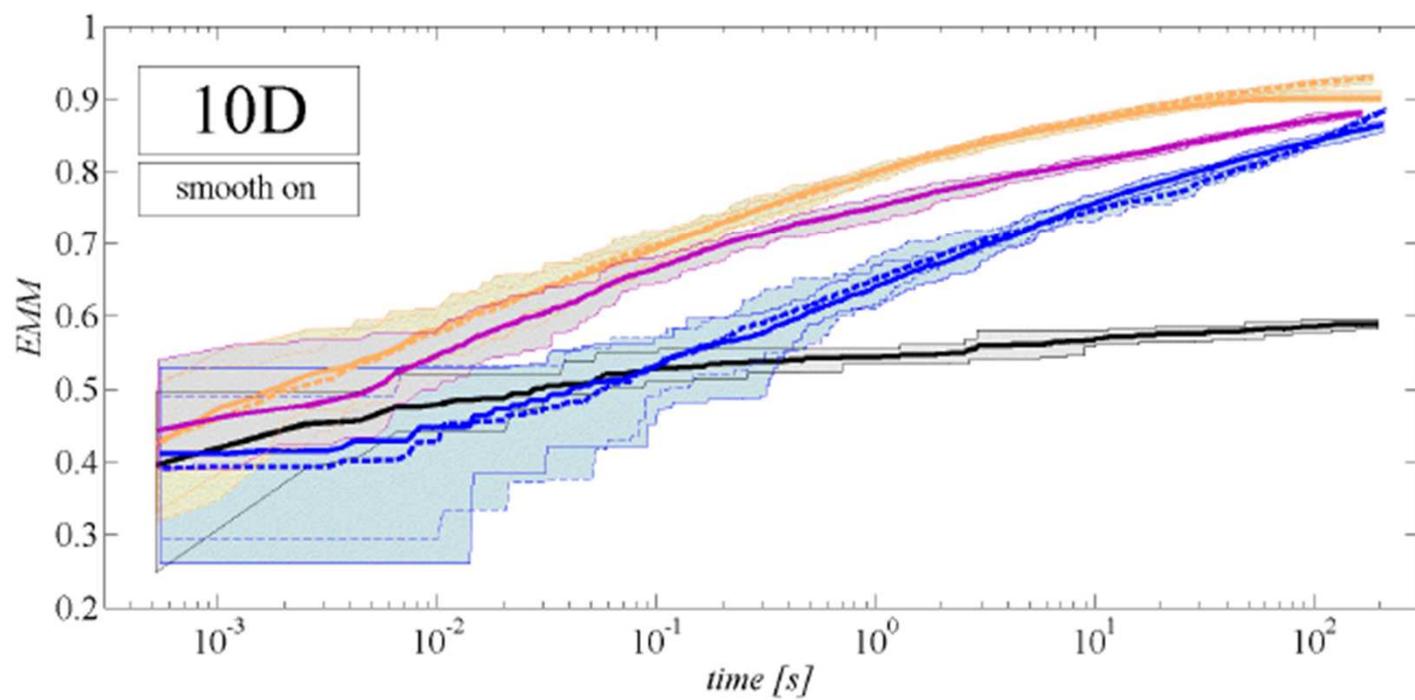
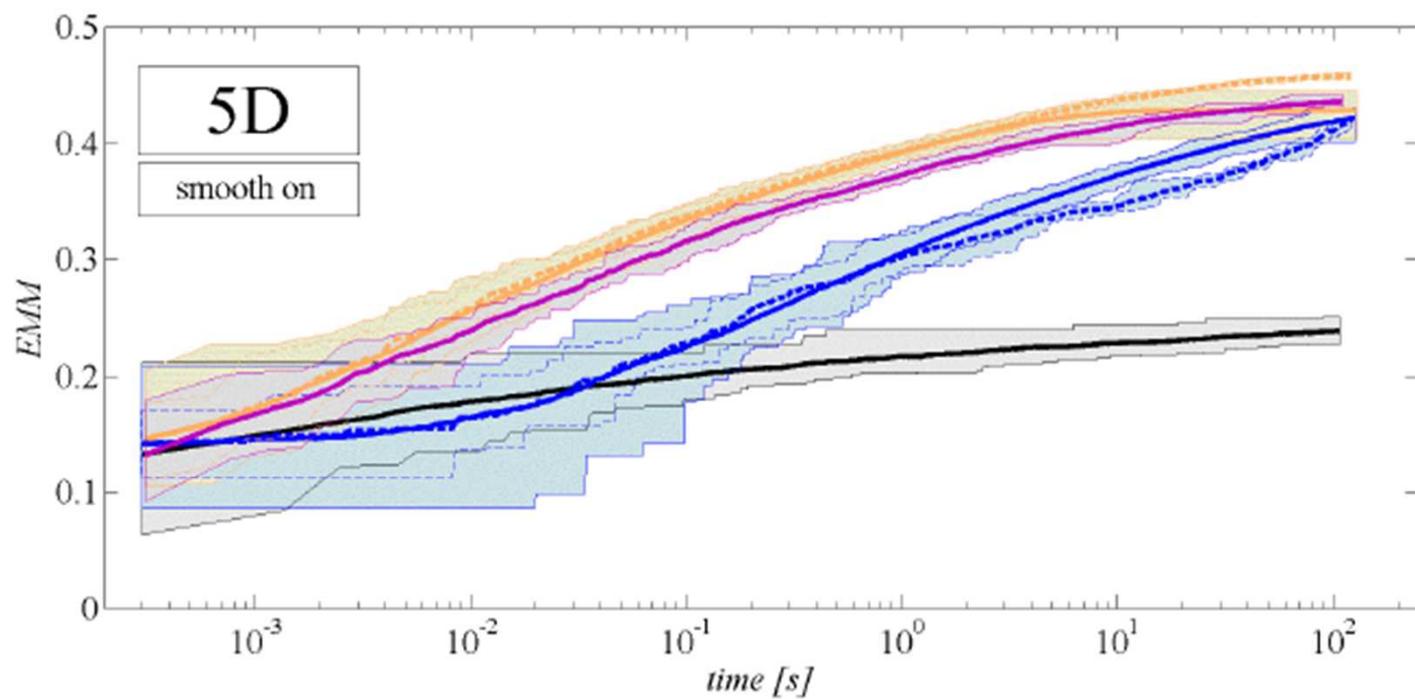
$$E^{\text{ML}_2} = \left(\frac{4}{3}\right)^k - \frac{2^{(1-k)}}{n} \sum_{d=1}^n \prod_{i=1}^k (3 - x_{di}^2) + \frac{1}{n^2} \sum_{d=1}^n \sum_{j=1}^n \prod_{i=1}^k [2 - \max(x_{di}, x_{ji})].$$

- D-optimalita (Dopt) [Kirsten Smith, 1918; M. Hofwing, 2010]

$$E^{\text{Dopt}} = - \det(\mathbf{Z}^T \mathbf{Z}), \quad \mathbf{Z} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{11}^2 & x_{12}^2 & x_{11}x_{12} \\ 1 & x_{21} & x_{22} & x_{21}^2 & x_{22}^2 & x_{21}x_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n1}^2 & x_{n2}^2 & x_{n1}x_{n2} \end{bmatrix}$$

Optimal Latin Hypercube Sampling – heuristic procedure plus Simulated annealing





Ortogonalita návrhu

- Číslo podmíněnosti (CN) [T. M. Cioppa, 2007]

$$E^{\text{CN}} = \text{cond}(\mathbf{X}^T \mathbf{X}) = \frac{\lambda_1}{\lambda_n}$$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}$$

- Pearsonův korelační koeficient (PMCC)

$$E^{\text{PMCC}} = \sqrt{\sum_{i=1}^k \sum_{j=i+1}^k c_{ij}^2} \quad c_{ij} = \frac{\text{Cov}(x_i, x_j)}{\sigma_{x_i} \sigma_{x_j}} = \frac{\sum_{a=1}^n (x_{a,i} - \bar{x}_i)(x_{a,j} - \bar{x}_j)}{\sqrt{\sum_{a=1}^n (x_{a,i} - \bar{x}_i)^2 \sum_{a=1}^n (x_{a,j} - \bar{x}_j)^2}}$$

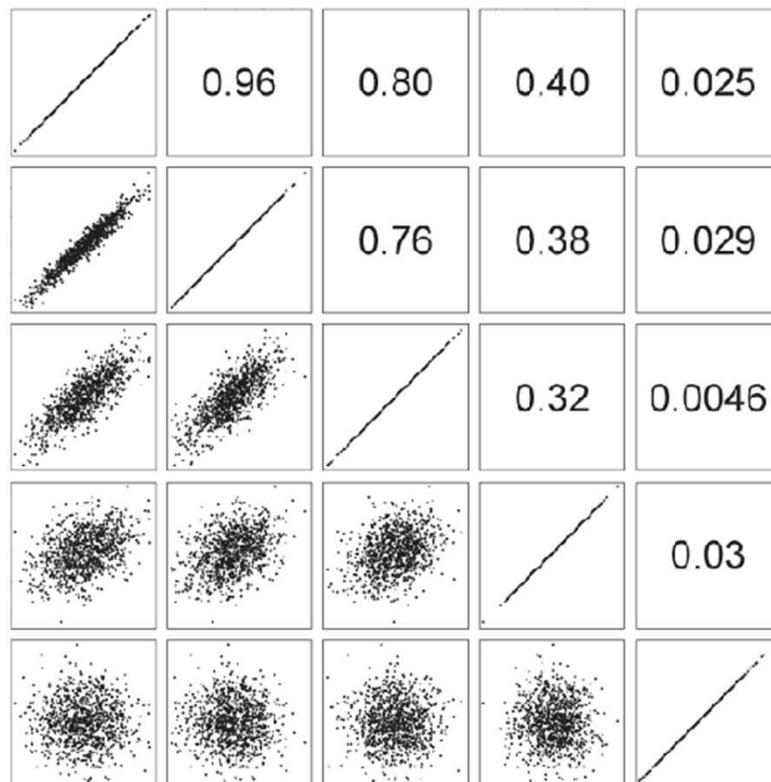
- Spearmanův koeficient pořadové korelace (SRCC)

$$E^{\text{SRCC}} = \sqrt{\sum_{i=1}^k \sum_{j=i+1}^k \rho_{ij}^2} \quad \rho_{ij} = 1 - \frac{6 \sum_{a=1}^n (r(x_{a,i}) - r(x_{a,j}))^2}{n(n^2 - 1)}$$

- Kendallův koeficient pořadové korelace (KRCC)

$$E^{\text{KRCC}} = \sqrt{\sum_{i=1}^k \sum_{j=i+1}^k \tau_{ij}^2} \quad \tau_{ij} = \frac{T_{c,ij} - T_{d,ij}}{n(n-1)/2}$$

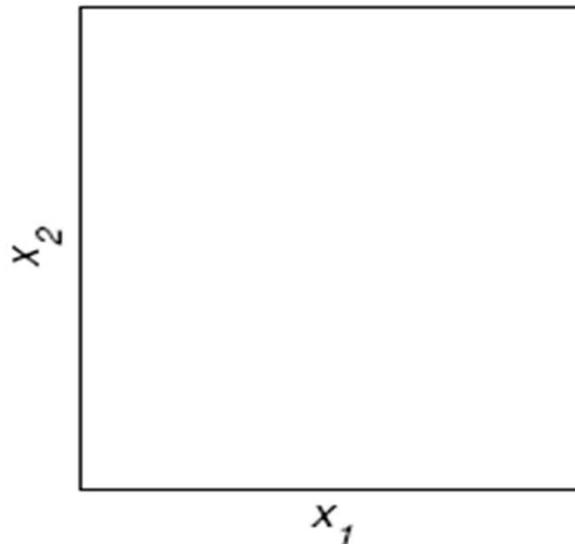
Předeepsaná korelační matice



Positive linear correlations between 1000 pairs of numbers.

Návrhová doména

- dána přítomností a podobou omezujících podmínek
- návrhová doména – dány pouze meze
→ hyperkrychle

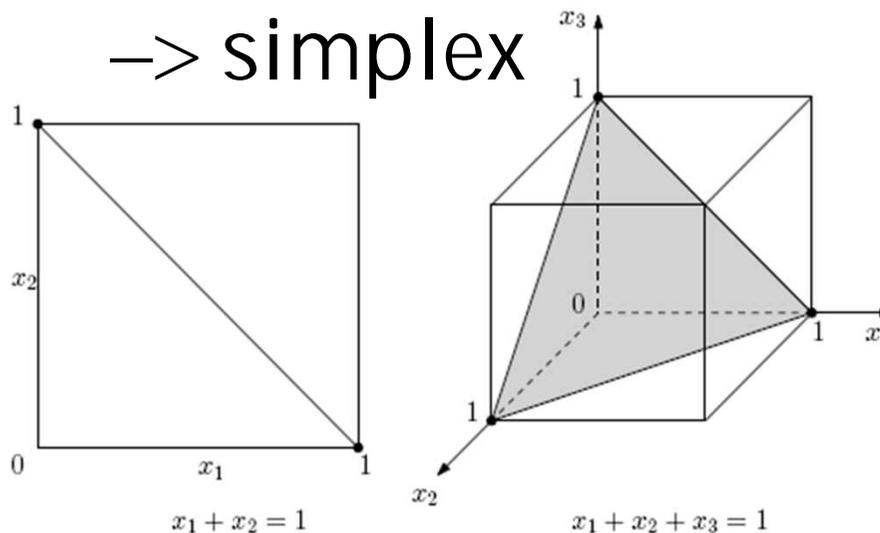


Návrhová doména

- dána přítomností a podobou omezujících podmínek
- návrh směsi (*mixture experiment*)

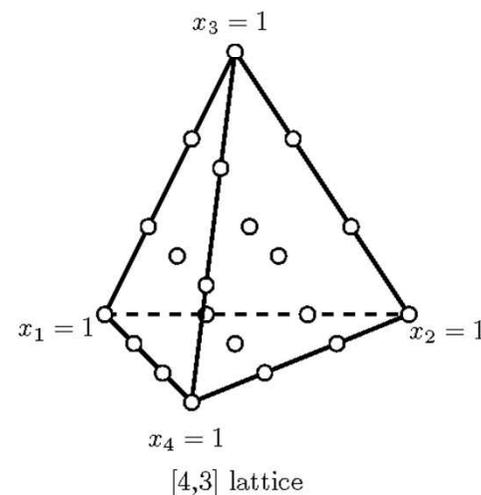
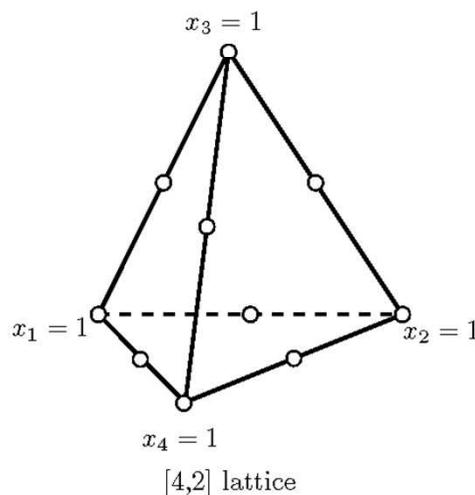
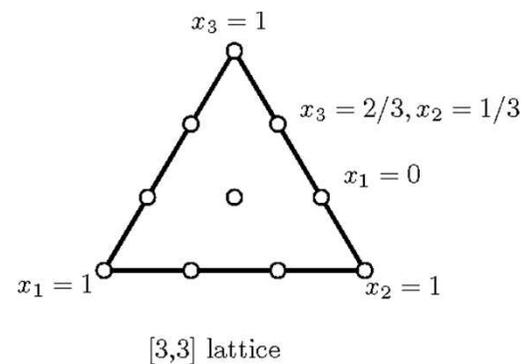
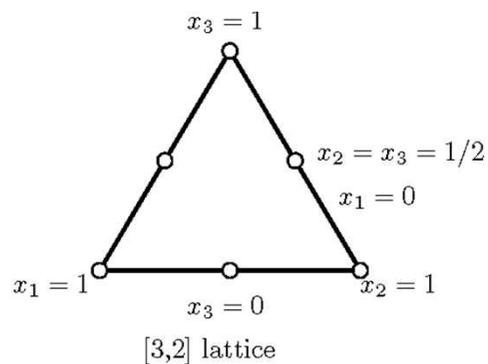
$$- \boxed{x_1 + x_2 + \dots + x_n = 1; 0 \leq x_i \leq 1; i = 1, \dots, n}$$

→ simplex

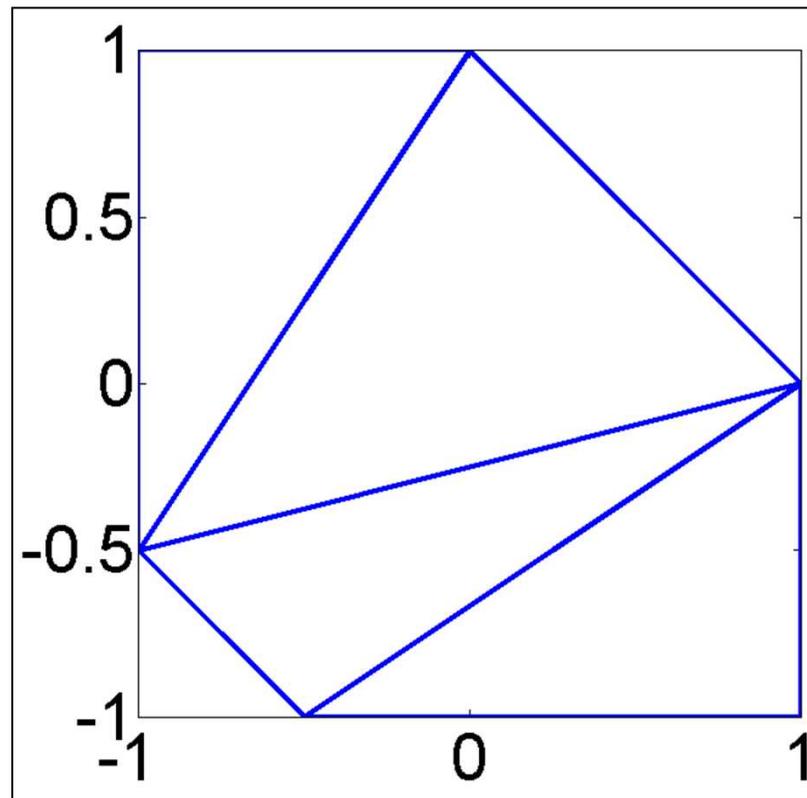


Metody tvorby DoE

- Klasické vzory



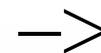
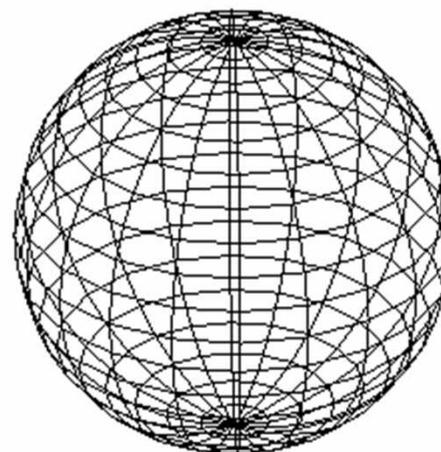
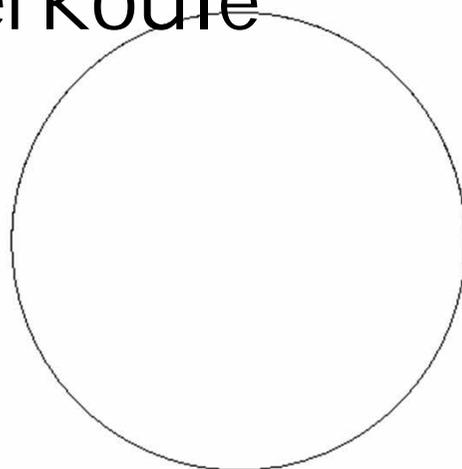
Distmesh tool



Návrhová doména

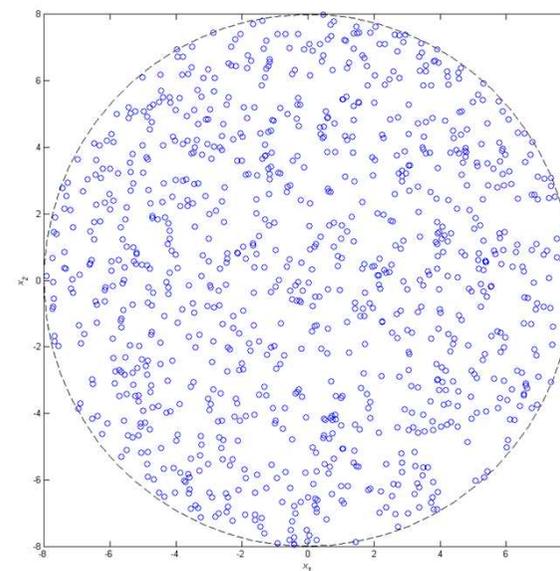
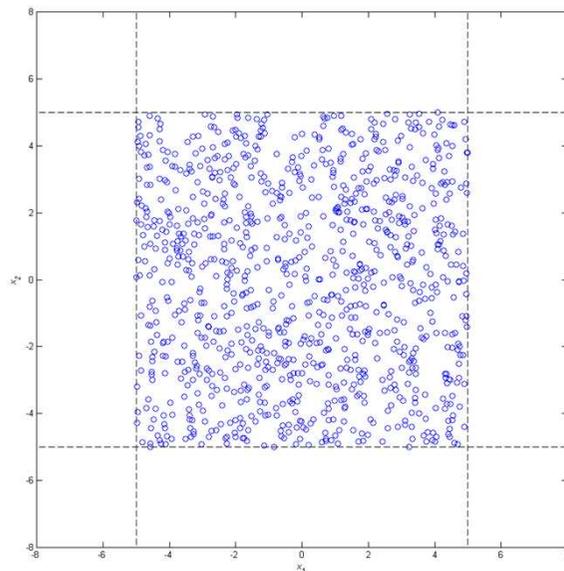
- dána přítomností a podobou omezujících podmínek
- omezená vzdálenost k danému bodu $\|x\|_2 \leq R$

hyperkoule



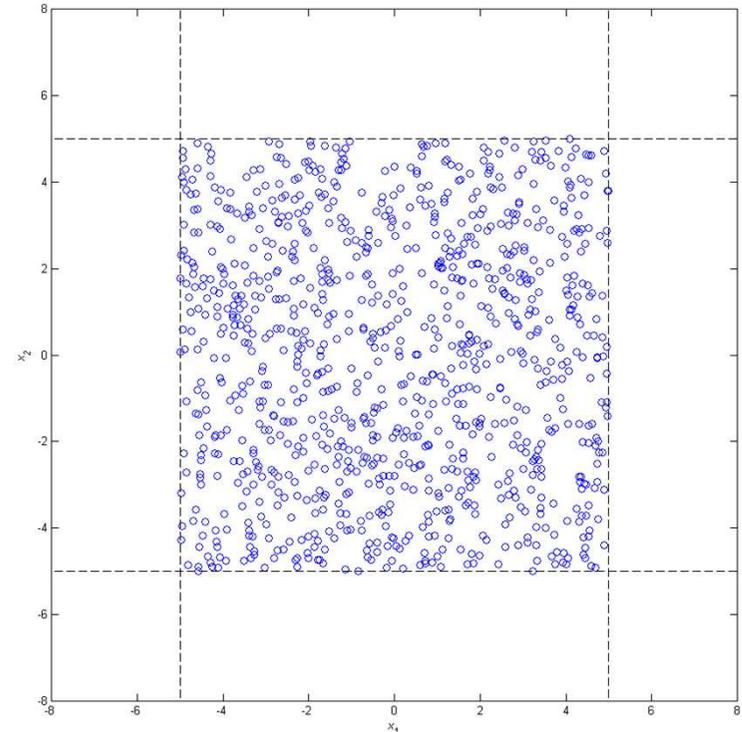
Poznámka: objem jednotkové hyperkrychle vs. jednotkové hyperkoule

Dimenze	Krychle	Koule
2	1	0.785
3	1	0.524
10	1	0.00249
20	1	$2.46 \cdot 10^{-8}$



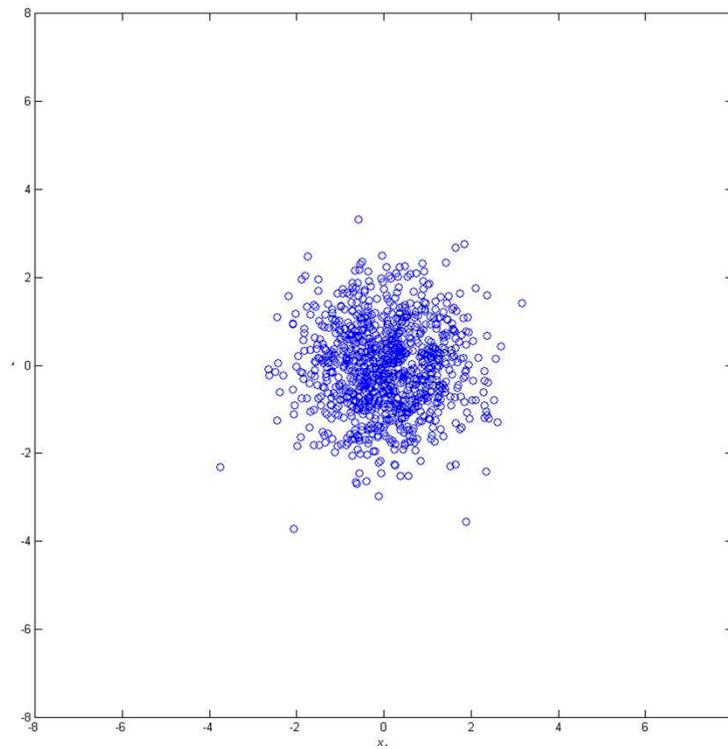
Sampling from hypercube

- ✓ Known methodology
- ✓ Fast and simple
- ✓ Enables adaptive sampling

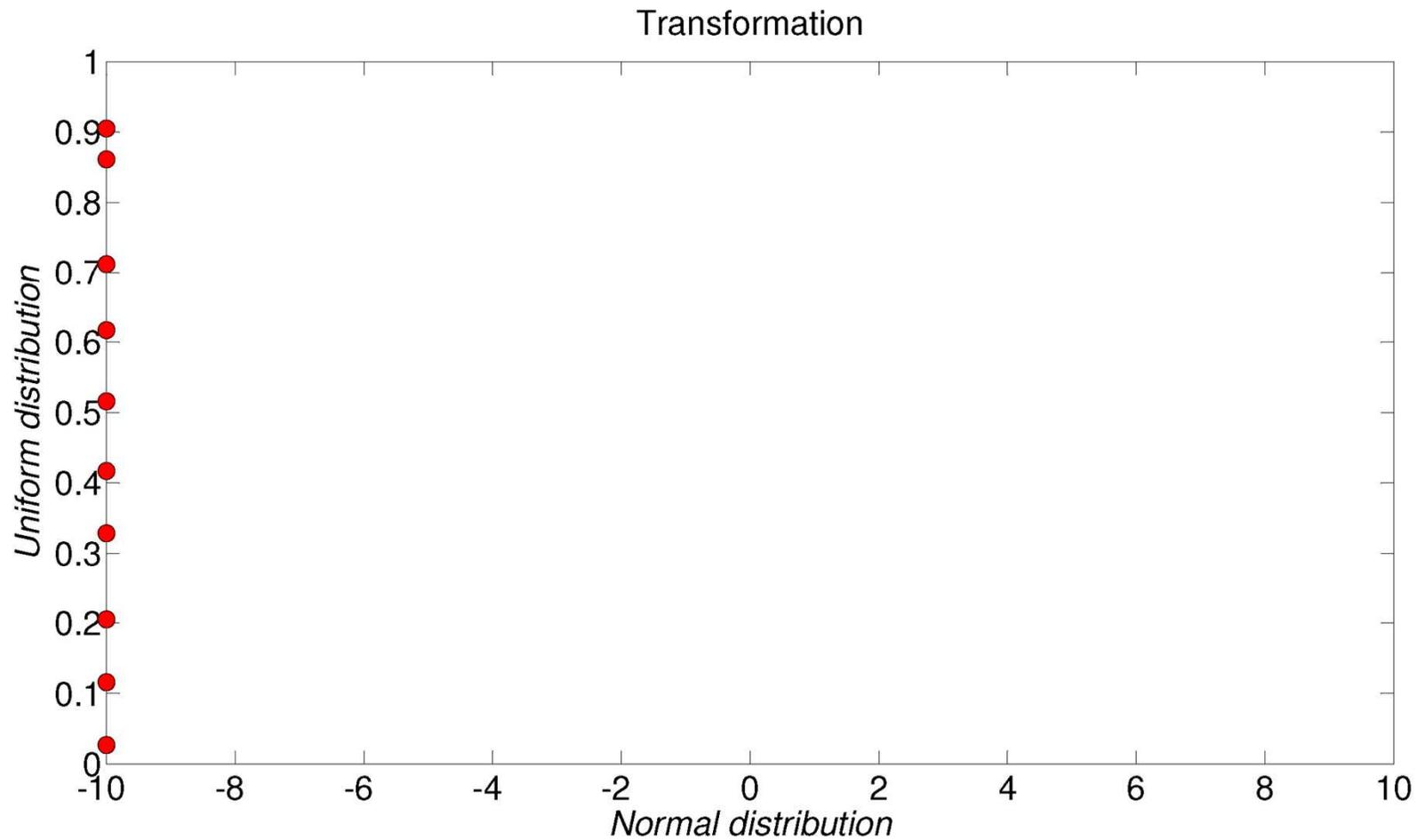


✗ Omits solutions outside bounds!

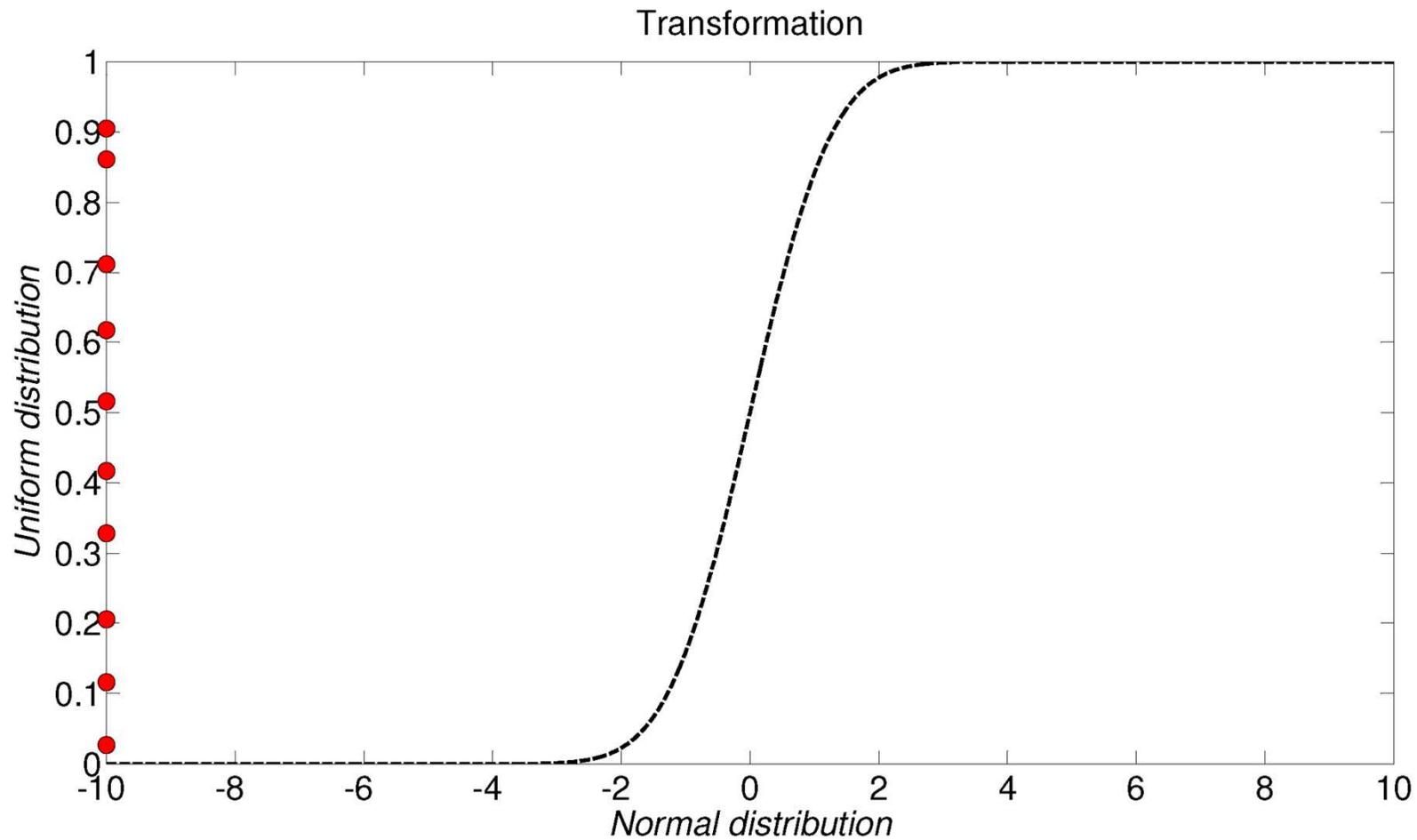
Sampling from prescribed distributions



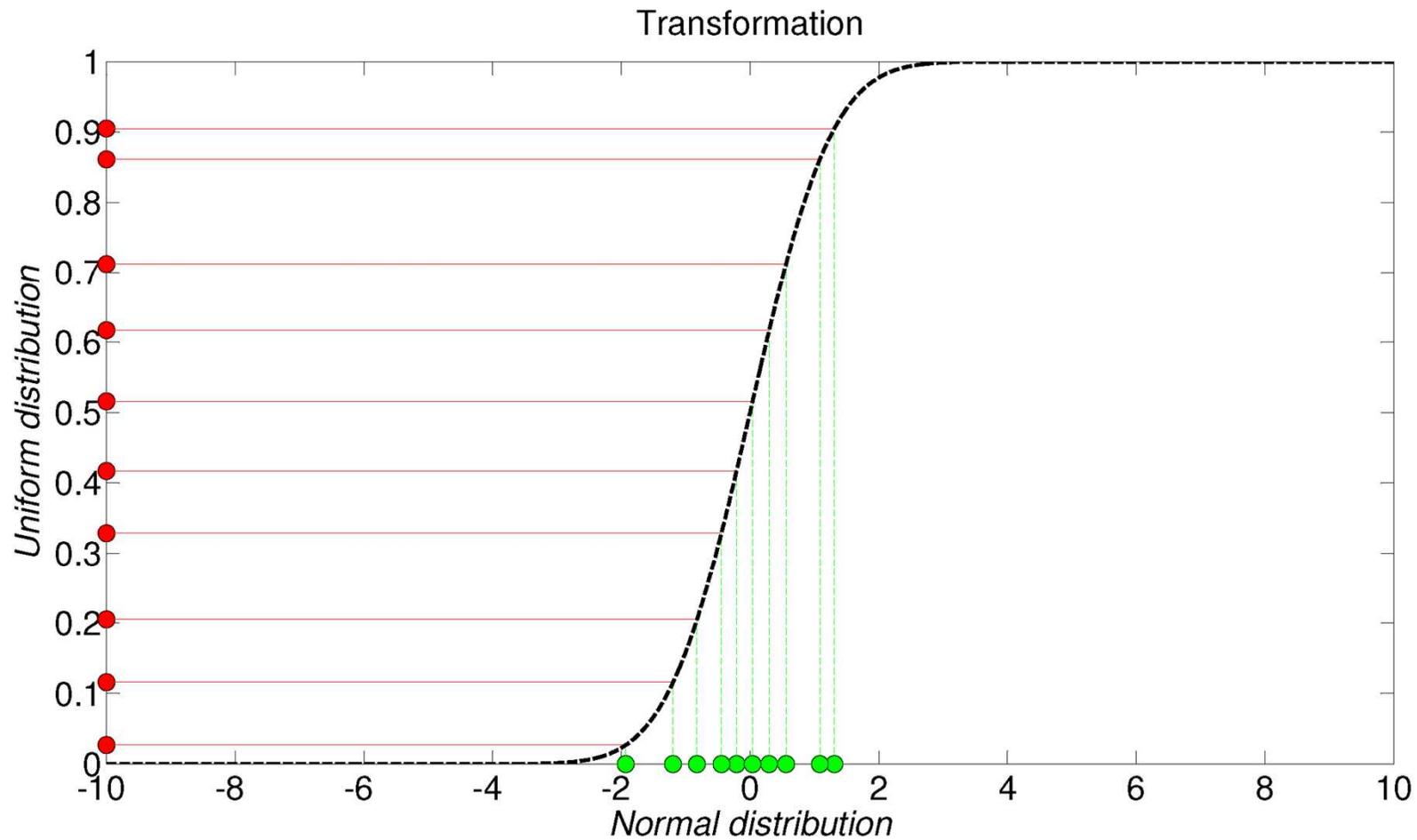
Transformation from uniform distribution



Transformation from uniform distribution

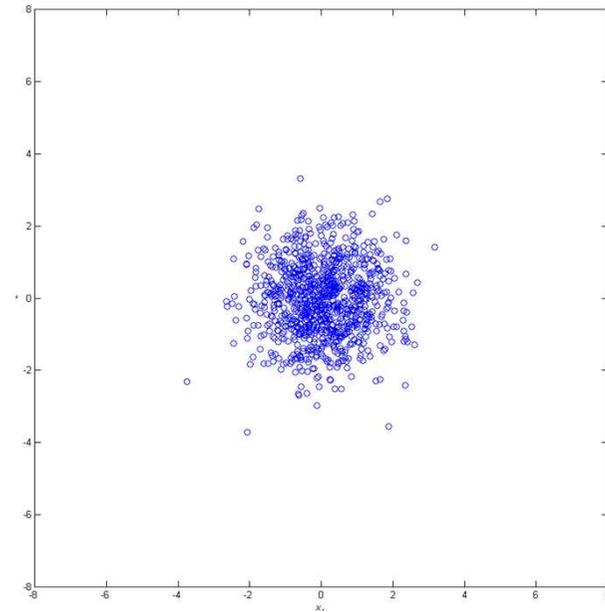


Transformation from uniform distribution

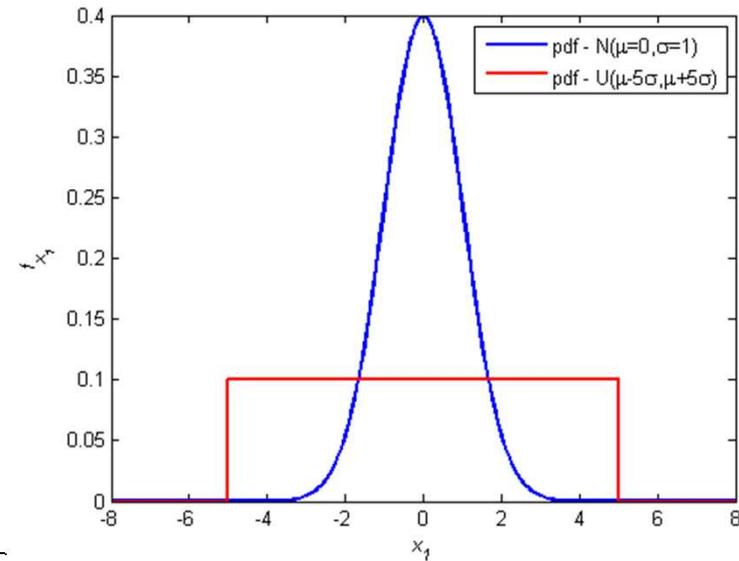
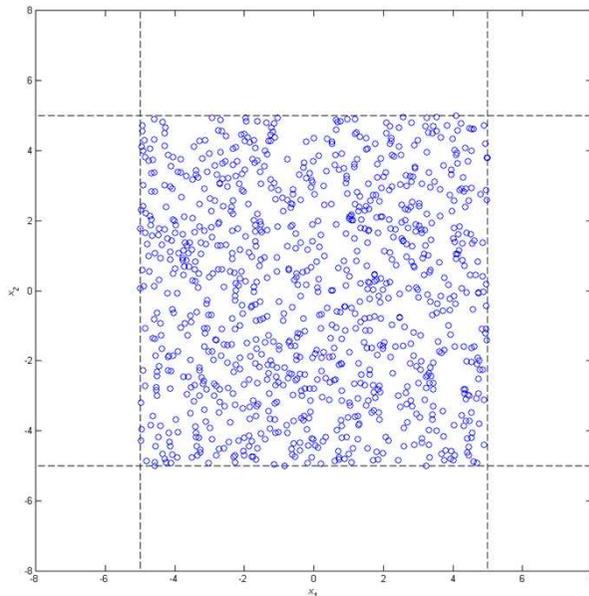
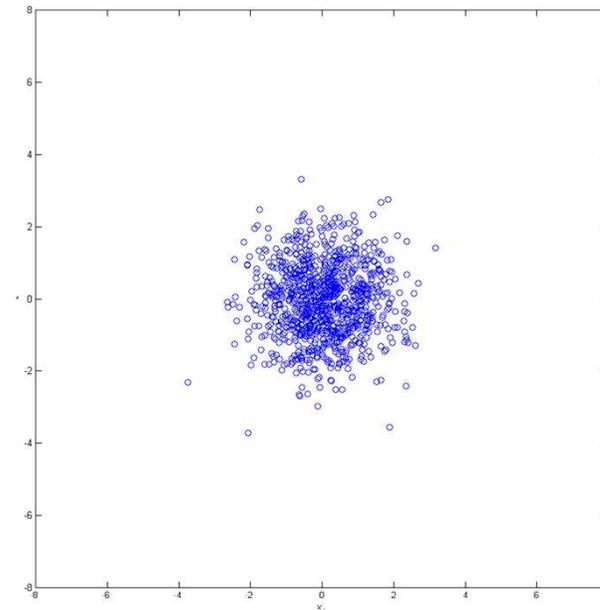


Sampling from prescribed distributions

- ✓ Known methodology
- ✗ Sampling around mean
- ✗ May miss failure region
- ✗ Problems with adaptive sampling



Hypercube vs. prescribed distribution?

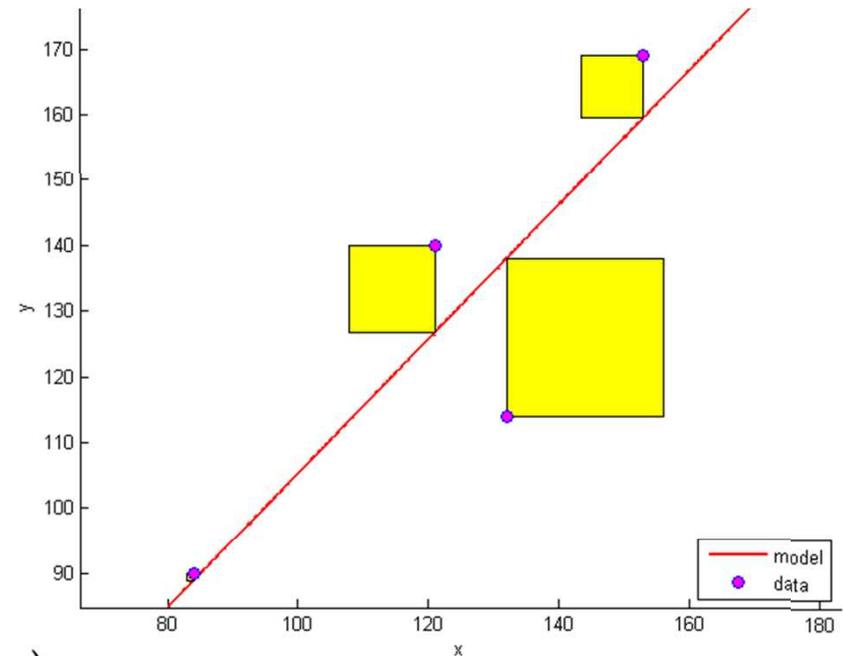


Lineární regrese

- Model: $\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i + \epsilon$
 $\underline{\hat{y}} = \underline{\Phi} \underline{\beta} + \underline{\epsilon}$

- x_i - vysvětlující veličina (regresor)
- \hat{y} - vysvětlovaná veličina (odezva regrese)
- ϵ - reziduální odchylka, $E(\epsilon) = 0$, $\text{Var}(\epsilon) = \sigma^2$
- i - pořadové číslo proměnné
- n - celkový počet proměnných
- β_i, β_0 nebo β - koeficienty
- Φ - matice návrhu experimentů, také zvana jako Valdermondova matice
- Odhad parametrů např. pomocí metody

neimenších čtverců: $\hat{\beta} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$



Minimální počet bodů DoE: $n + 1$

Příklad jednoduché lineární regrese

- Ve zkušebně chtěli zjistit, jaká je závislost mezi výkonností téhož výrobku za skutečných podmínek provozu. Za tímto účelem bylo vybráno 10 výrobků. U každého byla změřena jak výkonnost v testu X , tak výkonnost za skutečných podmínek provozu Y . Naměřené údaje jsou uvedeny v tabulce. Předpokládejte lineární závislost.

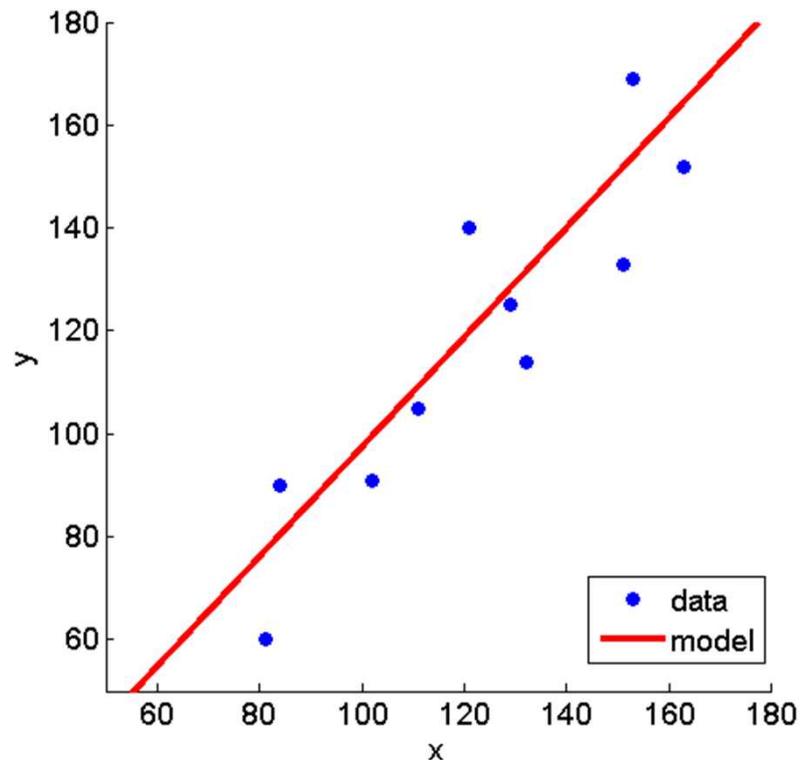
x	y
121	140
153	169
132	114
84	90
102	91
111	105
163	152
81	60
151	133
129	125

[Slovní zadání příkladu převzato ze skript prof. Jaruškové – Pravděpodobnost a matematická statistika, čísla změněna]

Příklad jednoduché lineární regrese

```
x = [121, 153, 132, 84, 102, 111, 163, 81, 151, 129];  
y = [140, 169, 114, 90, 91, 105, 152, 60, 133, 125];  
Psi = [ones(1,10) x]; % matice návrhu experimentů  
Beta_app = (Psi' * Psi) \ (Psi' * y')
```

```
Beta_app = -8.99340124551449  
          1.0341760492707
```



x	y
121	140
153	169
132	114
84	90
102	91
111	105
163	152
81	60
151	133
129	125

Model:

$$\hat{y} = -8.9934 + 1.0341x$$

Regrese zahrnující interakci veličin

- Model: $\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \beta_{ij} x_i x_j$

- x - vysvětlující veličina (regresor)
- \hat{y} - vysvětlovaná veličina (odezva regrese)
- n - počet vysvětlujících veličin
- β_0, β_i - koeficienty
- Příklad pro $n = 3$: $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3$

Minimální počet bodů DoE: $1 + \frac{n(n+1)}{2}$

- Odhad parametrů např. pomocí metody

nejmenších čtverců $\underline{\hat{\beta}} = \left(\underline{\Phi}^T \underline{\Phi} \right)^{-1} \underline{\Phi}^T \underline{y}$

- $\underline{\Phi}$ - matice návrhu experimentu (tzv. Vandermondova matice)
- \underline{y} - odezva původního modelu

Kvadratická regrese

- Model: $\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \sum_{j=i}^n \beta_{ij} x_i x_j$

- x - vysvětlující veličina (regresor)
- \hat{y} - vysvětlovaná veličina (odezva regrese)
- n - počet vysvětlujících veličin
- $\beta_0, \beta_i, \beta_{ij}$ - koeficienty
- Příklad pro $n = 3$: $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{11} x_1^2 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{22} x_2^2 + \beta_{23} x_2 x_3 + \beta_{33} x_3^2$

Minimální počet bodů

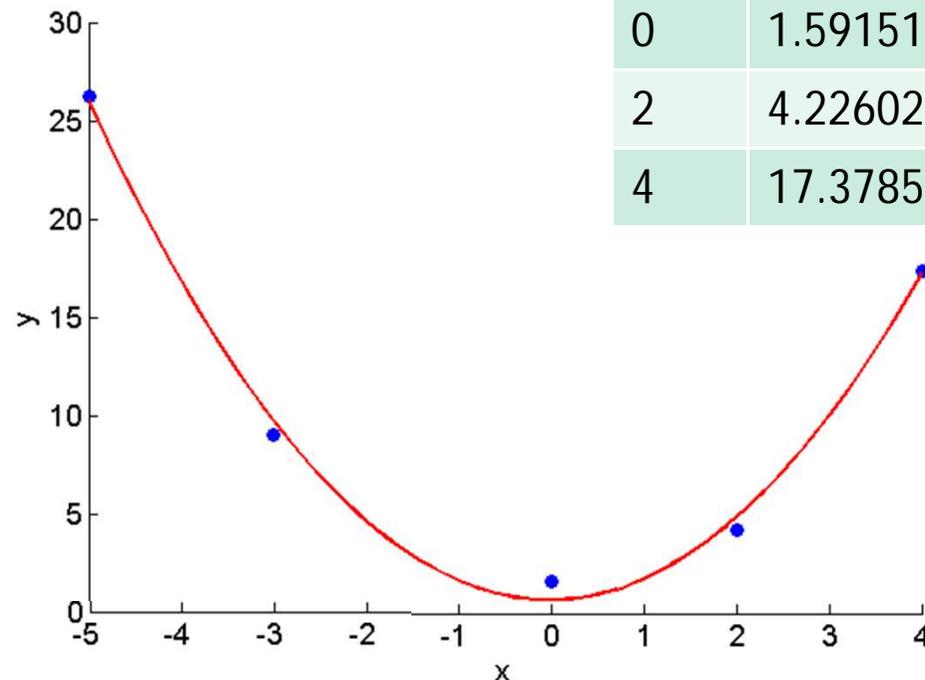
$$\text{DoE: } 1 + n + \frac{n(n+1)}{2}$$

Příklad jednoduché kvadratické regrese

```
x = [-5, -3, 0, 2, 4];  
y = [26.274046, 9.0840786, 1.5915102, 4.2260288, 17.378550];  
Psi = [ones(1,5) x' x'.^2]; % matice návrhu experimentů  
Beta_app = (Psi'*Psi)\(Psi'*y')
```

```
Beta_app = 0.679526827730912  
           0.047343996311551  
           1.02317162054266
```

x	y
-5	26.274046
-3	9.0840786
0	1.5915102
2	4.2260288
4	17.378550



Model:

$$\hat{y} = 0,67953 + 0,04734x + 1,02317x^2$$

Polynomiální regrese pro 1 vysvětlující vel.

- Model: $\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_m x^m = \sum_{i=0}^m \beta_i x^i$

- x - vysvětlující veličina (regresor)
- \hat{y} - vysvětlovaná veličina (odezva regrese)
- m - celkový stupeň polynomu
- β_0, β_i - koeficienty
- Odhad parametrů např. pomocí metody nejmenších čtverců $\underline{\hat{\beta}} = \left(\underline{\Phi}^T \underline{\Phi} \right)^{-1} \underline{\Phi}^T \underline{y}$
- $\underline{\Phi}$ - matice návrhu experimentu (tzv. Vandermondova matice)
- \underline{y} - odezva původního modelu

Polynomiální regrese

- Pascalův trojúhelník pro kompletní polynomy ve 2D (pro 2 vysvětlující veličiny)

			1			Konstantní	Min. 1 bod
		x		y		Lineární	Min. 3 body
	x ²		xy		y ²	Kvadratický	Min. 6 bodů
x ³		x ² y		xy ²		y ³	Min. 10 bodů
x ⁴	x ³ y		x ² y ²		xy ³	y ⁴	Polynom 4. řádu Min. 15 bodů
						Polynom m. řádu	Min. (m+1)!

- Pro více neznámých obdobně, počet bodů DoE roste se stupněm polynomů a zároveň s počtem vysvětlujících veličin

$$\text{Minimální počet bodů DoE: } \frac{(m+n)!}{m! n!}$$

Kriging

normálně rozdělená náhodná
chyba s nenulovou kovariancí

- (neznámá) funkce: $y(\mathbf{x}) = f(\mathbf{x}) + \mathbf{Z}(\mathbf{x})$

$$E(Z_i) = 0, \quad V(Z_i) = \sigma^2, \quad \text{Cov}[\mathbf{Z}(\mathbf{x}^i)\mathbf{Z}(\mathbf{x}^j)] = \sigma^2 \mathbf{R}([R(\mathbf{x}^i \mathbf{x}^j)])$$

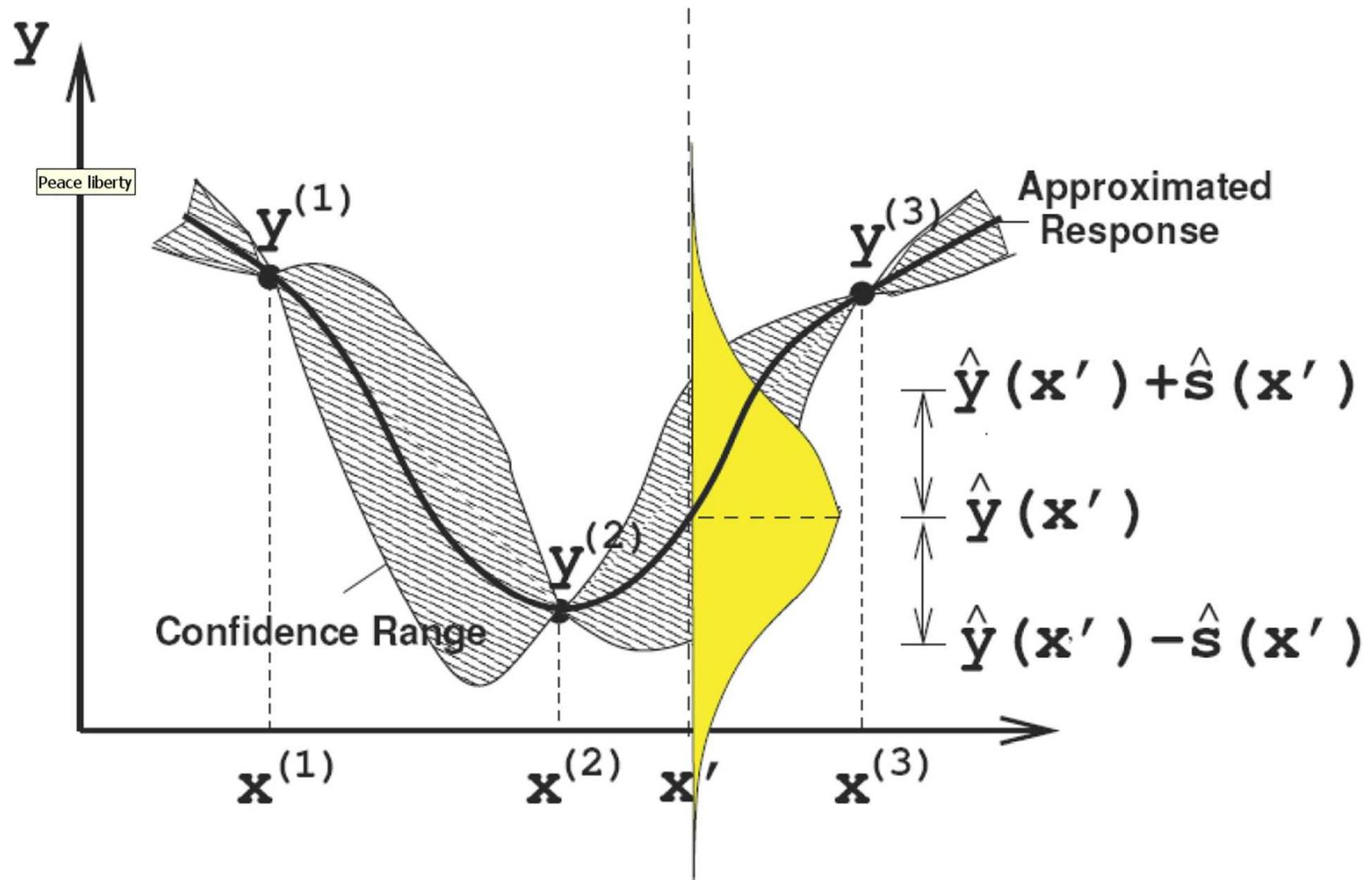
$$\mathbf{R}(\mathbf{x}^i, \mathbf{x}^j) = \exp\left[\sum_{k=1}^N \theta_k |x_k^i - x_k^j|^2\right]$$

- aproximace:

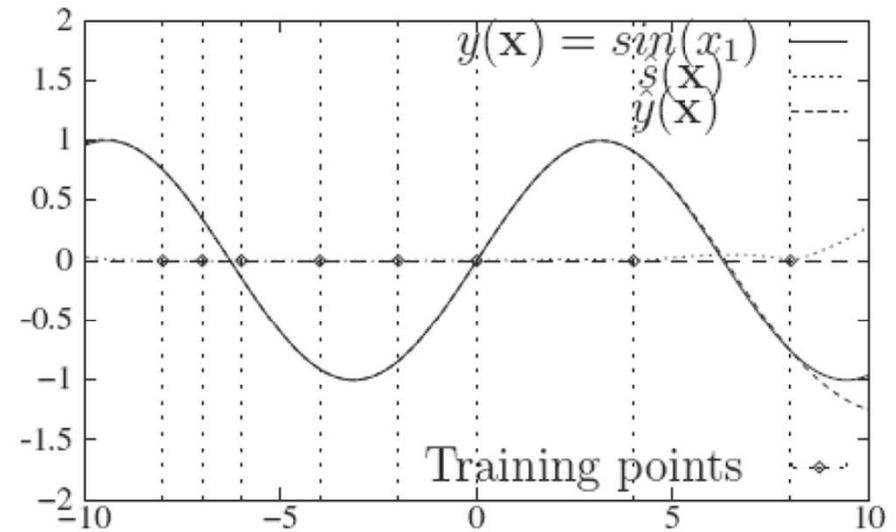
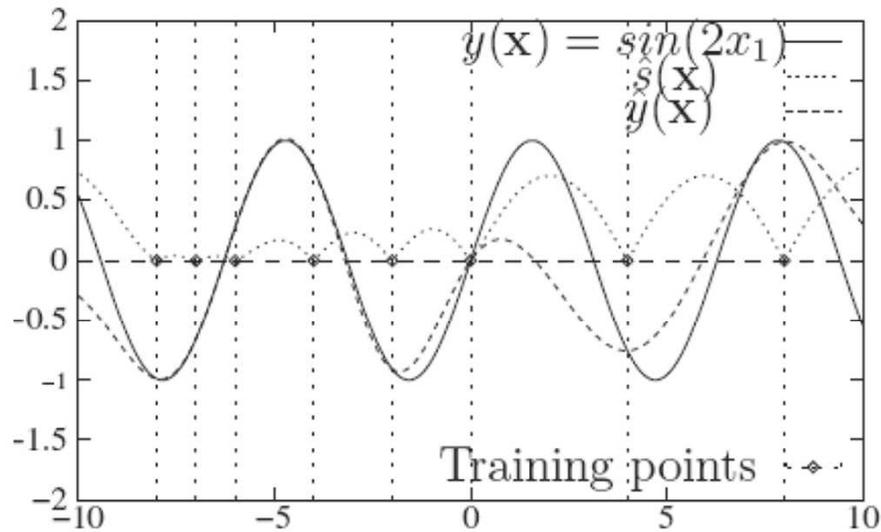
$$\hat{y}(\mathbf{x}) = \beta + (\mathbf{y} - \mathbf{1}\beta)^T \mathbf{R}^{-1} \mathbf{r}(\mathbf{x})$$

$$\hat{V}(\mathbf{x}) = \sigma^2 \cdot \left[\mathbf{1} - \mathbf{r}(\mathbf{x})^T \cdot \mathbf{R}^{-1} \cdot \mathbf{z}(\mathbf{x}) + \frac{(\mathbf{1} - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}))^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right]$$

Kriging



Kriging



- predikované hodnoty jsou přesné ve známých bodech
- predikce chyby jsou velké na hrubé ploše a malé na hladké ploše
- predikce chyby rostou se vzdáleností od známých bodů

RBFN (Radial-Basis Function Network)

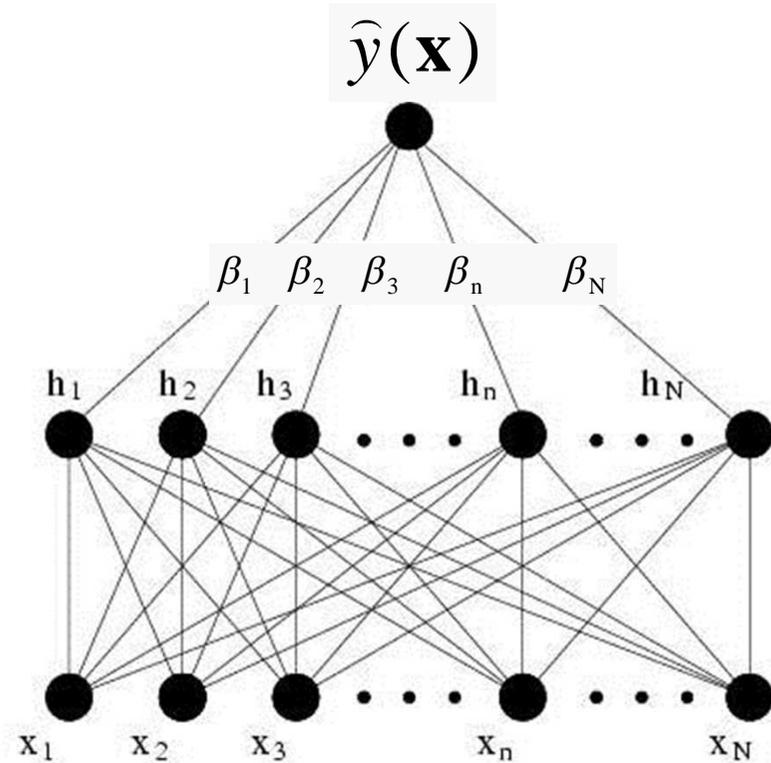
(Sítě s radiální bází)

– Funkce je aproximována:

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^N \beta_i h_i(\mathbf{x})$$

Bázové funkce : $h_i(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/r}$

Váhy β_i vypočteny z rovnosti funkčních hodnot funkce a její aproximace v N bodech x_i ... vede na soustavu lineárních rovnic!



Trénovací body

Radiální báze

- Model: $\hat{y}(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(\|\mathbf{x} - \mathbf{c}^i\|)$
- \hat{y} - odezva meta-modelu
- x - vysvětlující veličina
- w - váhové koeficienty
- $\boldsymbol{\psi}$ - vektor bazových funkcí
- $\|\cdot\|$ - Euklidovská norma
- c - centra bazových funkcí
- n_c - počet bodů DoE prokládaných meta-modelem

Radiální báze

- Model: $\hat{y}(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(\|\mathbf{x} - \mathbf{c}^i\|)$
- Bázové funkce:
 - Fixní
 - $\psi(r) = r, \psi(r) = r^3, \psi(r) = r^2 \ln(r), \dots$
 - Parametrické
 - $\psi(r) = e^{\frac{-r^2}{2\sigma^2}}, \psi(r) = \sqrt{r^2 + \sigma^2}, \psi(r) = \frac{1}{\sqrt{r^2 + \sigma^2}}, \psi(r) = \frac{1}{r^2 + \sigma^2}, \psi(r) = e^{-\sigma r}, \dots$

Radiální báze

- Model: $\hat{y}(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(\|\mathbf{x} - \mathbf{x}^i\|)$
- Nejjednodušší případ – centra bazových funkcí se ztotožní s body návrhu experimentu
- Odhad parametrů
 - $\mathbf{w} = \boldsymbol{\Psi}^{-1} \mathbf{y}$
 - $\Psi_{i,j} = \psi(\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|)$
 $c^{(i)} = \mathbf{x}^{(i)}$
- Je-li soustava špatně podmíněná, pak se dá k diagonále matice $\boldsymbol{\Psi}$ přičíst malé číslo

Ilustrativní příklad radiálních bází

- Původní model: $y = x^2$
- Meta-model: radiální báze s lineárními fixními bázemi
 $\psi(r) = r$
- Body návrhu experimentu $x = [4, 2, 6]$
- Odezvy původního modelu $y = [16, 4, 36]$
- Centra bází: $x^{(1)} = 4, x^{(2)} = 2, x^{(3)} = 6$

- $$\Psi = \begin{bmatrix} \psi(\|4 - 4\|) & \psi(\|4 - 2\|) & \psi(\|4 - 6\|) \\ \psi(\|2 - 4\|) & \psi(\|2 - 2\|) & \psi(\|2 - 6\|) \\ \psi(\|6 - 4\|) & \psi(\|6 - 2\|) & \psi(\|6 - 6\|) \end{bmatrix} = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 4 \\ 2 & 4 & 0 \end{bmatrix}$$

- $$\Psi^{-1} = \begin{bmatrix} -0,5 & 0,25 & 0,25 \\ 0,25 & -0,125 & 0,125 \\ 0,25 & 0,125 & -0,125 \end{bmatrix}$$

Ilustrativní příklad radiálních bází

- $\mathbf{w} = \Psi^{-1}\mathbf{y} = \begin{bmatrix} -0,5 & 0,25 & 0,25 \\ 0,25 & -0,125 & 0,125 \\ 0,25 & 0,125 & -0,125 \end{bmatrix} \begin{bmatrix} 16 \\ 4 \\ 36 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 0 \end{bmatrix}$
- Centra bází: $c^{(1)} = 4, c^{(2)} = 2, c^{(3)} = 6$
- $\hat{y}(3) = \mathbf{w}^T \boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(\|x - c^i\|) = 2 \cdot \psi(\|3 - 4\|) + 8 \cdot \psi(\|3 - 2\|) + 0 \cdot \psi(\|3 - 6\|) = 2 \cdot 1 + 8 \cdot 1 + 0 \cdot 3 = 10$
- $y(3) = 3^2 = 9$

Kombinace aproximačních a interpolačních meta-modelů

- Vhodné využít výhody obou typů meta-modelů
- Např. kombinace lineární regrese a radiálních bází
 - Data \mathbf{X} , odezvy \mathbf{y}
 - Výpočet regresních koeficientů pro LR a odezev $\hat{\mathbf{y}}_{LR}$
 - Model LR: $y_{LR} = \beta_0 + \sum_{i=1}^n \beta_i x_i + \epsilon$
 - $\underline{\hat{\boldsymbol{\beta}}} = \left(\underline{\boldsymbol{\Phi}}^T \underline{\boldsymbol{\Phi}} \right)^{-1} \underline{\boldsymbol{\Phi}}^T \underline{\mathbf{y}}$
 - odezvy $\hat{y}_{LR} = \beta_0 + \sum_{i=1}^n \beta_i x_i$
 - Odečtení hodnot odezvy původního modelu a odezvy LR a získání chyby $\boldsymbol{\varepsilon} = \mathbf{y} - \hat{\mathbf{y}}_{LR}$

Kombinace aproximačních a interpolačních meta-modelů

- Aproximace chyby $\boldsymbol{\varepsilon}$ pomocí radiálních bází

• Model:

$$\hat{y}_{RBF}(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(\|\mathbf{x} - \mathbf{c}^i\|)$$

• Váhové koeficienty $\mathbf{w} = \boldsymbol{\Psi}^{-1} \boldsymbol{\varepsilon}$

- Celková aproximace: $\hat{y}(\mathbf{x}) = \hat{y}_{LR}(\mathbf{x}) + \hat{y}_{RBF}(\mathbf{x})$

Problém

- Aproximační nástroje obvykle vybrány tak, aby byly značně jednodušší než optimalizovaný problém
- Aproximace pouze na základě výsledků z DoE obvykle nepopisuje problém dostatečně => nutnost iteračního postupu

Algoritmus ①

1. DoE vytvoří nová řešení, ohodnotí je na P
2. Přidání nových řešení do M
3. Nastavení (aproximace) M
4. Optimalizace M – získání nových řešení
5. Ohodnocení nových řešení na P
6. Pokud ne *konvergence*, návrat do bodu 2

Algoritmus ②

1. Inicializace optimalizačního algoritmu OA – obvykle DoE, ohodnocení pomocí P, nastavení M
2. Vytvoření nových řešení OA
3. M vytvoří „odhad“ hodnot nových řešení
4. OA si podle „odhadu“ M vybere, která řešení spočítá pomocí P
5. Řešení ohodnocená P vložena do M, nastavení M
6. Pokud OA iteruje, tak pokračuje bodem 2

Porovnání ① a ②

- Použití algoritmů závisí na důvěře v schopnosti meta-modelu věrně popsat tvar problému:
 - Algoritmus ① je založen na úplné důvěře (meta-model řídí optimalizaci)
 - Algoritmus ② je naopak založen na značné nedůvěře (optimalizace pokud chce, tak použije meta-model)

RBFN (Radial-Basis Function Network)

(Sítě s radiální bází)

– Funkce je aproximována:

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^N \beta_i h_i(\mathbf{x})$$

Bázové funkce : $h_i(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/r}$

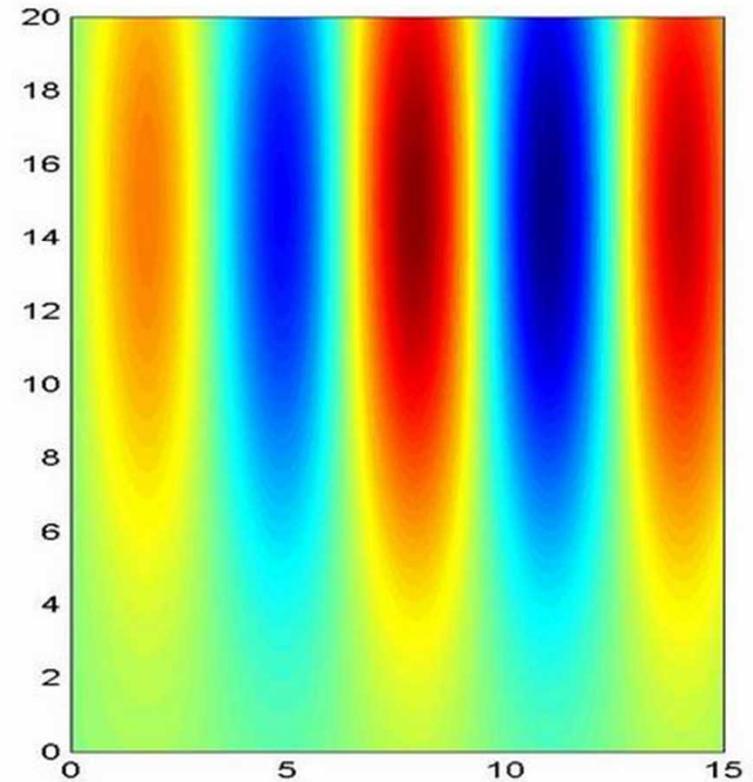
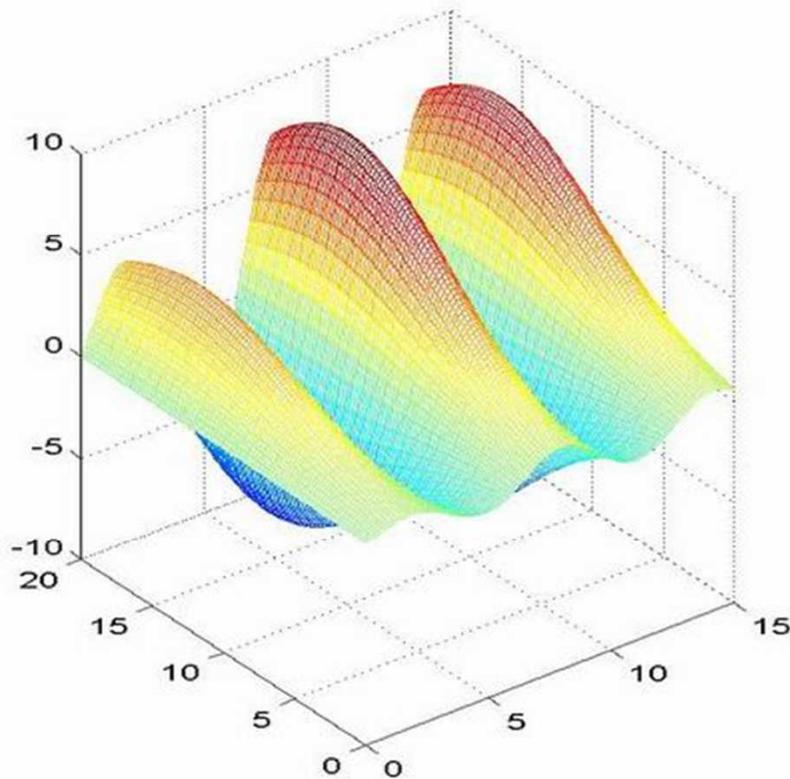
Váhy β_i vypočteny z rovnosti funkčních hodnot funkce a její aproximace v N bodech x_i

- Na aproximaci nalezeno optimum pomocí GA
- Přidání dalších trénovacích bodů
 - Optimum nalezené GA
 - Náhodný bod
 - Bod ve směru lepšího ze dvou posledních optim získaných GA (jednoduchý gradient)

RBFN

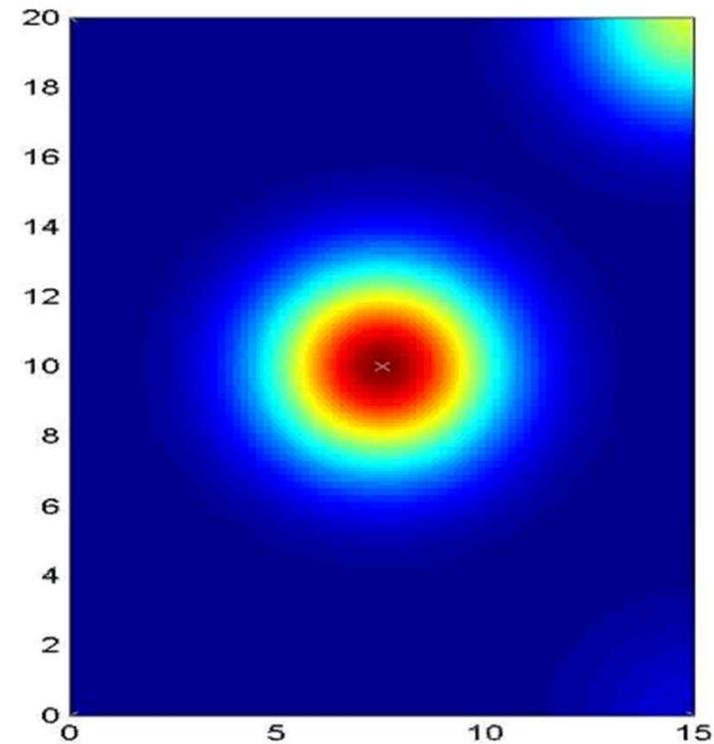
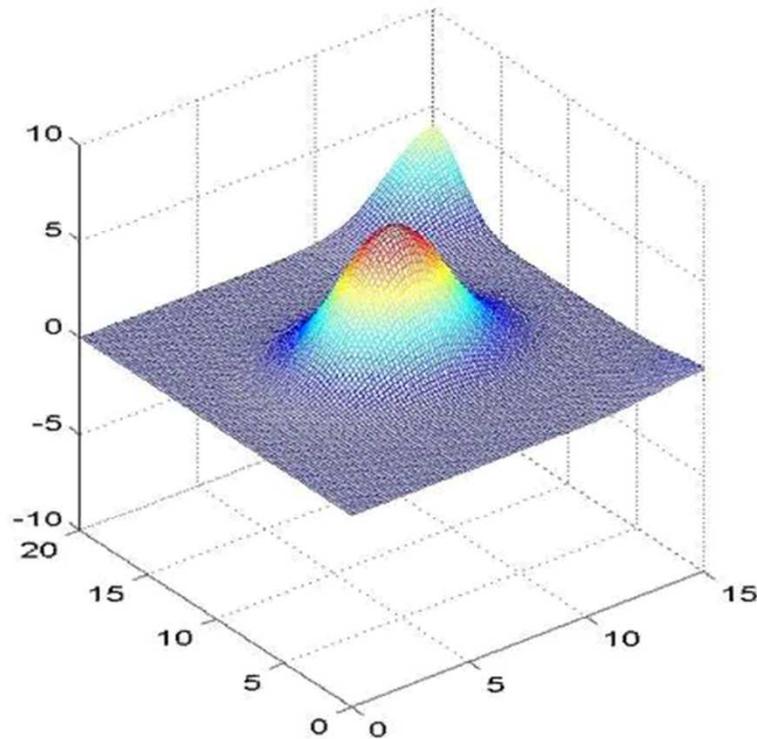
– Jednoduchý testovací příklad

$$ex1(x, y) = 10e^{(-0.01(x-10)^2 - 0.01(y-15)^2)} \sin(x)$$



RBFN

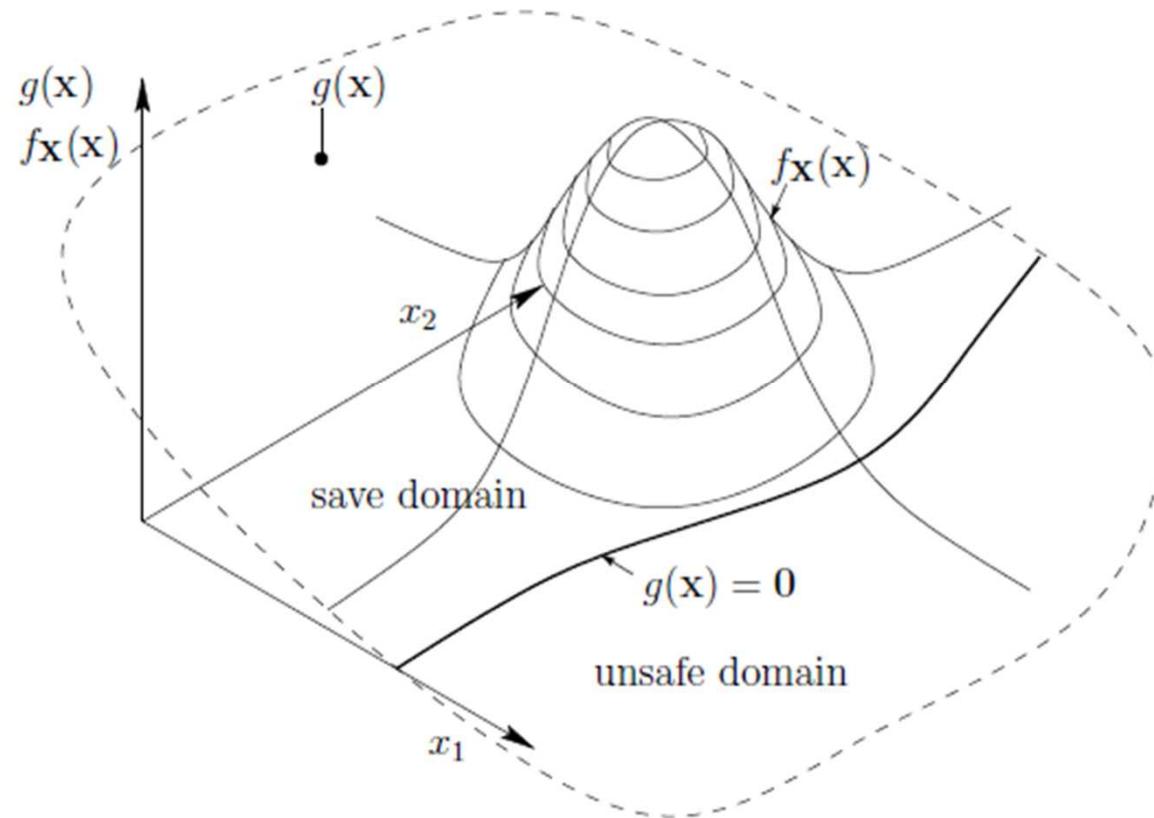
- Řešení úlohy ex1 s využitím algoritmu GRADE jako optimalizační metody



RBFN

Test function	<i>N</i>	SADE		GRADE		GRADE+CERAF		GRADE+RBFN	
		SR %	ANFC	SR %	ANFC	SR %	ANFC	SR %	ANFC
F1	1	100.0	61	100.0	61	100.0	60	100.0	23
F3	1	100.0	87	100.0	97	100.0	94	96.7	159
Branin	2	100.0	668	100.0	371	100.0	368	100.0	43
Camelback	2	100.0	306	100.0	223	100.0	222	100.0	61
Goldprice	2	100.0	634	100.0	360	100.0	358	11.6	472
PShubert1	2	100.0	1518	100.0	5501	100.0	1844	2.1	466
PShubert2	2	100.0	1043	100.0	1403	100.0	970	2.5	530
Quartic	2	100.0	534	100.0	341	100.0	339	100.0	77
Shubert	2	100.0	682	100.0	649	100.0	654	18.0	506
Hartman1	3	100.0	478	100.0	319	100.0	320	99.9	63
Shekel1	4	100.0	7719	100.0	33776	100.0	3434	0.0	-
Shekel2	4	100.0	4595	100.0	13522	100.0	2638	0.0	-
Shekel3	4	100.0	4127	100.0	10857	100.0	2650	0.0	-
Hartman2	6	71.2	57935	60.8	165622	100.0	10284	97.7	163
Hosc45	10	100.0	7759	100.0	2265	100.0	2274	-	-
Brown1	20	91.1	160515	100.0	209214	100.0	195250	-	-
Brown3	20	100.0	60554	100.0	36339	100.0	36429	-	-
F5n	20	94.4	26786	99.8	7197	100.0	7259	-	-
F10n	20	66.4	227577	70.3	90687	98.2	289702	-	-
F15n	20	97.5	48533	99.4	23358	100.0	24894	-	-

Adaptive sampling around LSF



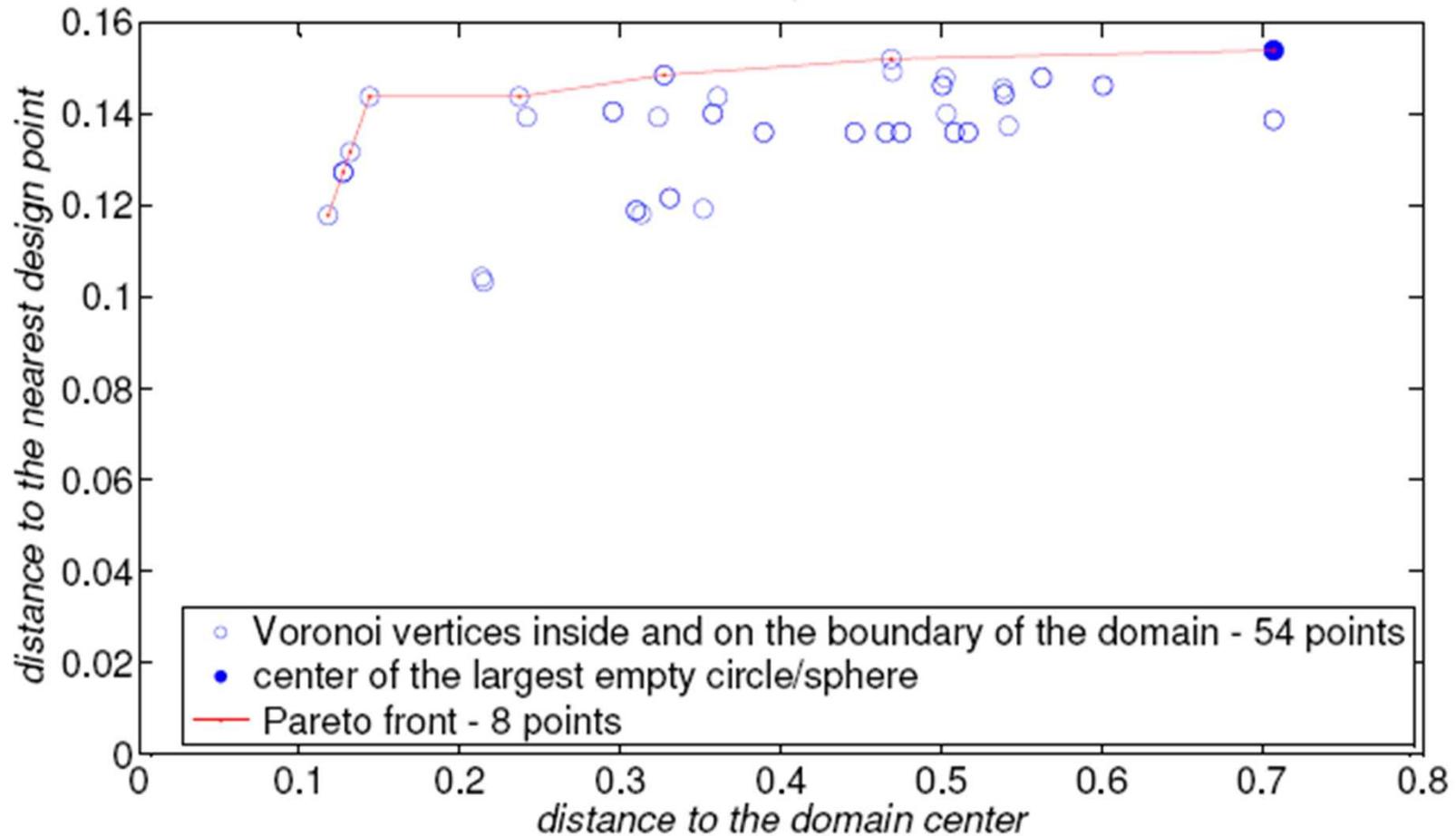
[Roos, 2006]

Surrogate Model

- Appropriate number of sampling points is needed
- Adaptive updating procedure
 - Multi-objective optimization problem
 - Maximization of the nearest distance of the added point from already sampled points (like miniMax metric)
 - To be as close as possible to the approximate limit state surface

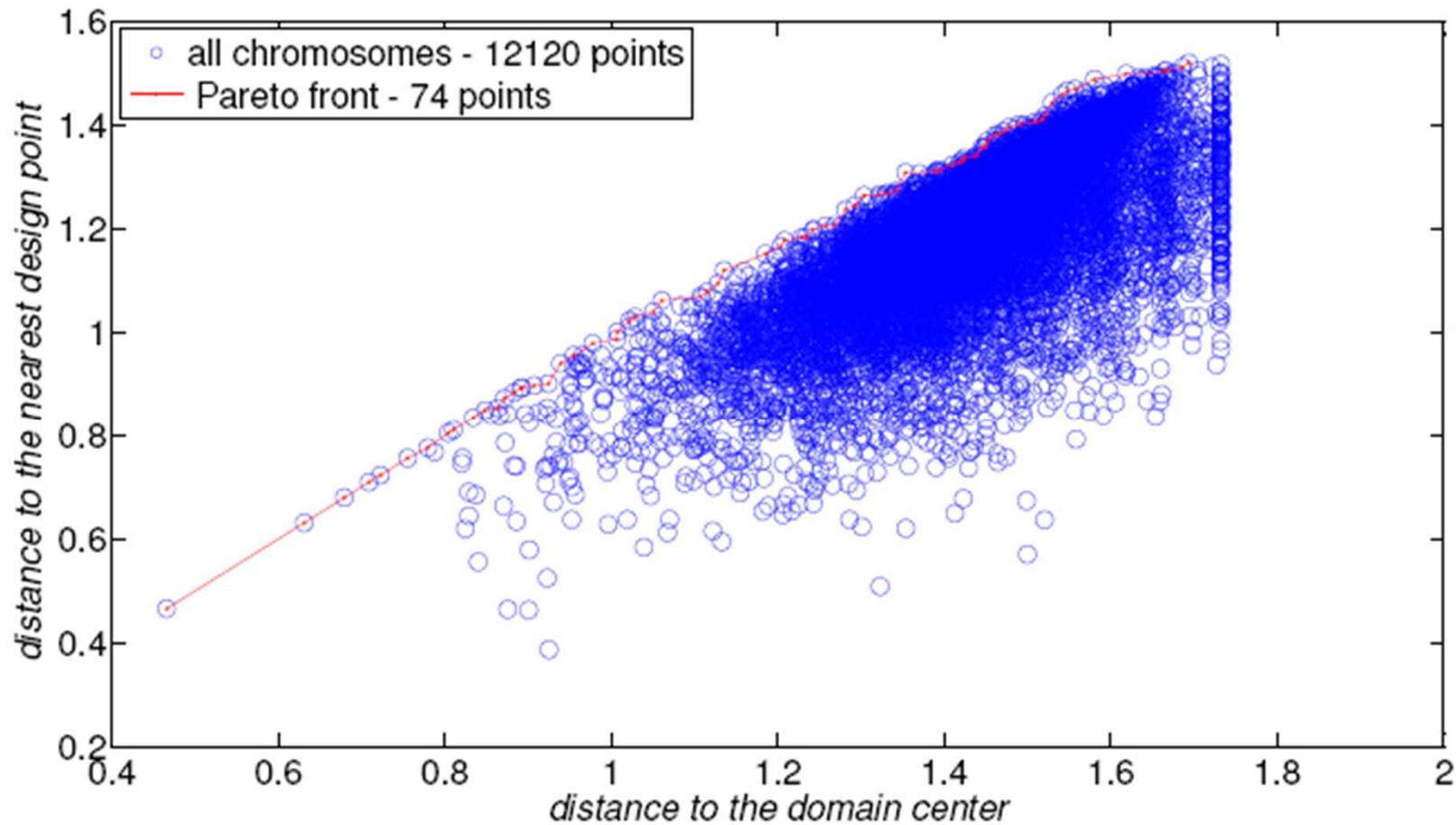
Multi-objective adaptive sampling

2D, 27 points



Multi-objective adaptive sampling

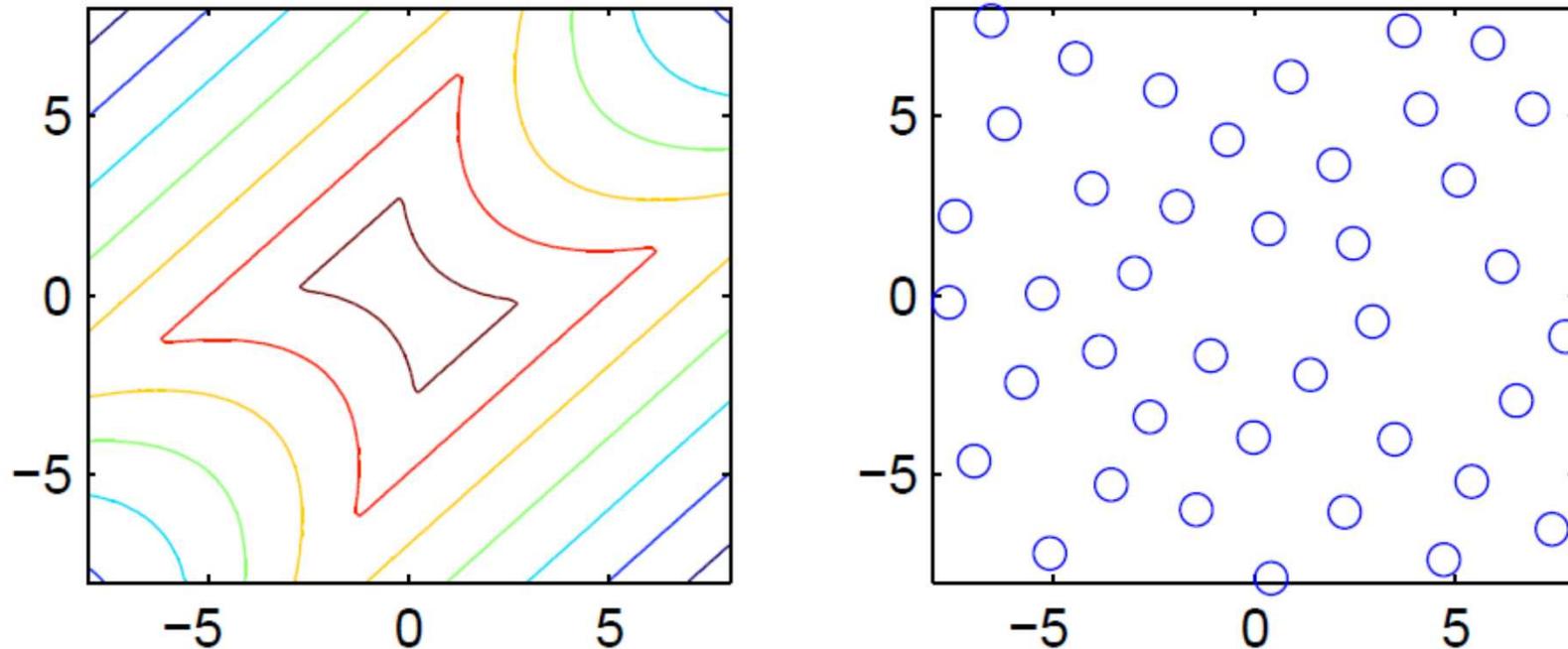
12D, 65 points



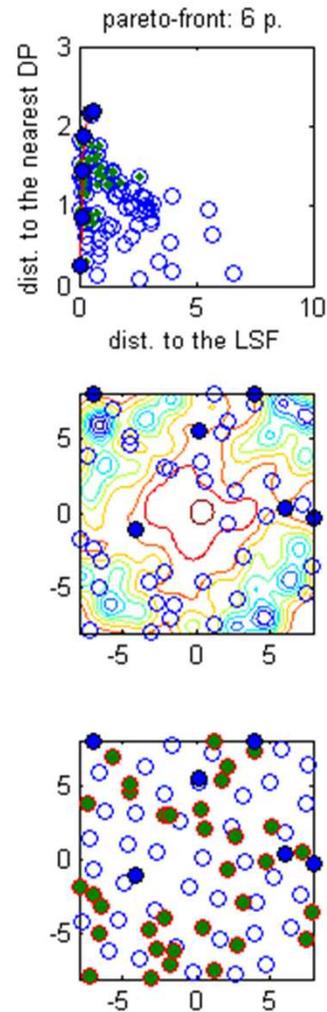
Implemented Meta-Models

- RBFN from Matlab
 - Neural Network based
- CTU implementation of RBFN
 - with different polynomial regression parts
- Kriging
 - DACE toolbox in Matlab
 - with different polynomial regression parts
 - with regression part found by Genetic Programming

Adaptive update of meta-model



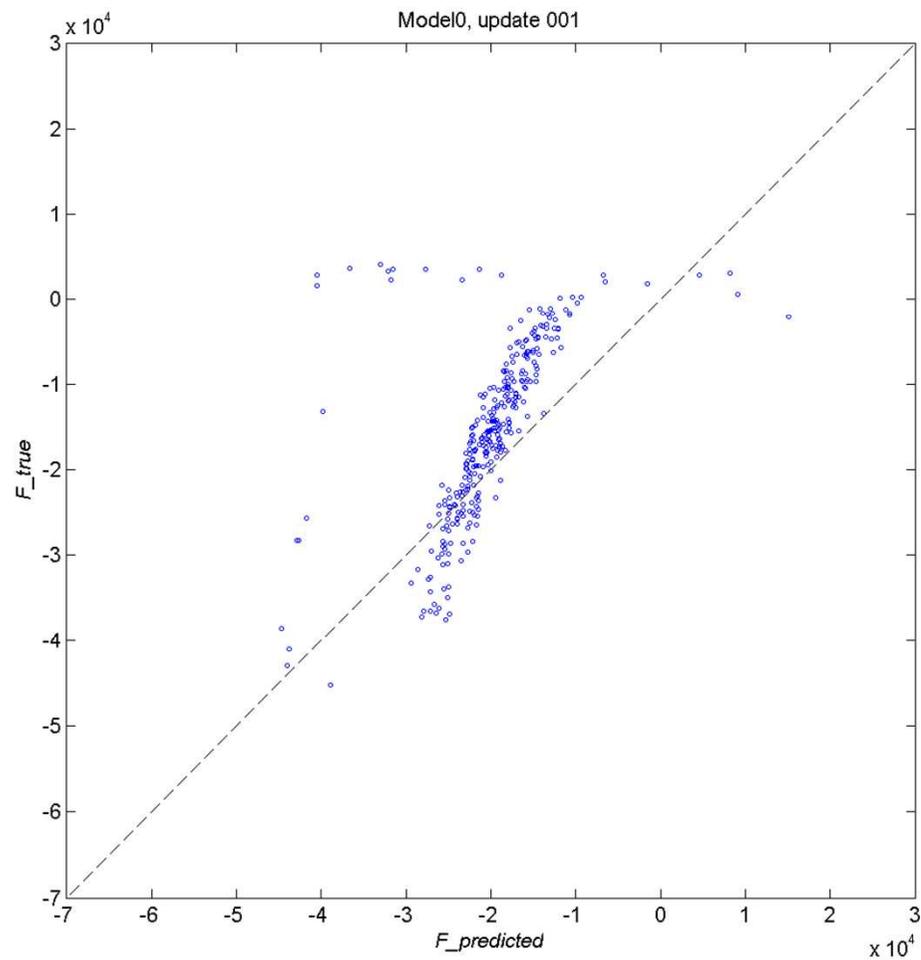
Contours of the example (left) and starting DoE (right). Note that the red contour is for $F(x) = 0$.



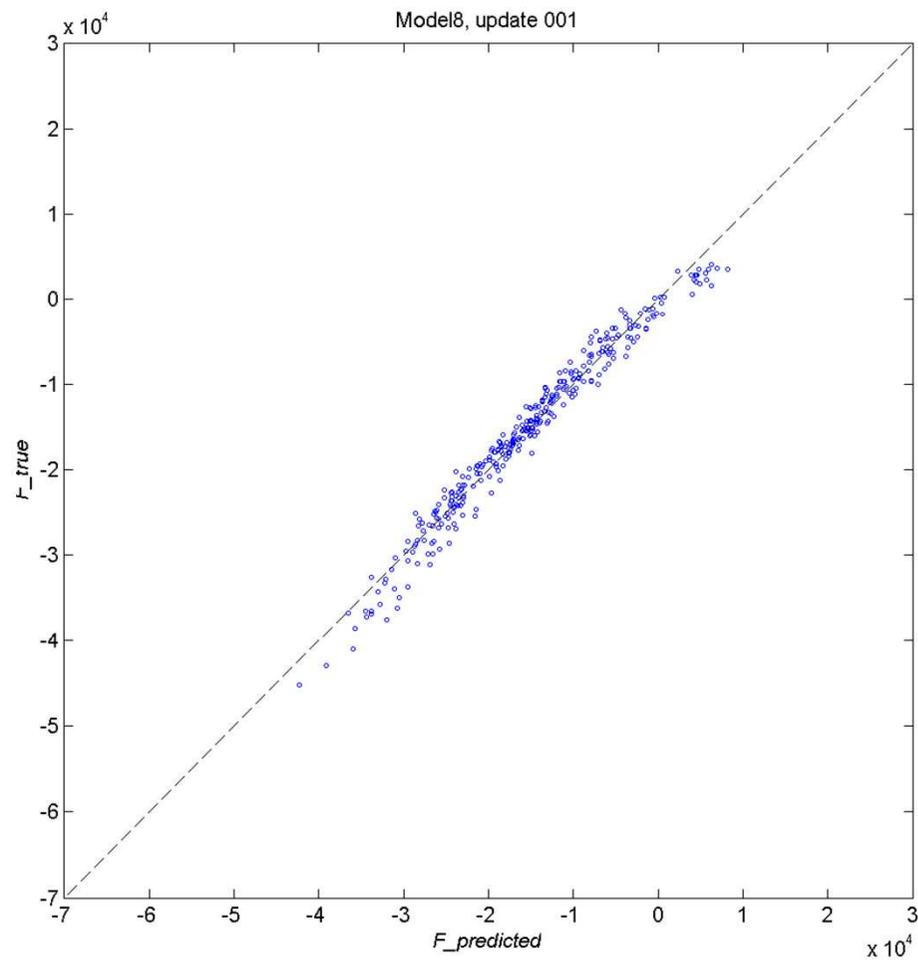
Pareto front (top), contours of the problem with DoEs (middle) and DoEs' points (bottom).
 Key: Red – added and computed solutions, Blue – points that were too close to other Pareto front points,
 Green – the remaining points of population and Blue empty points – the original DoE.

Quality of a metamodel

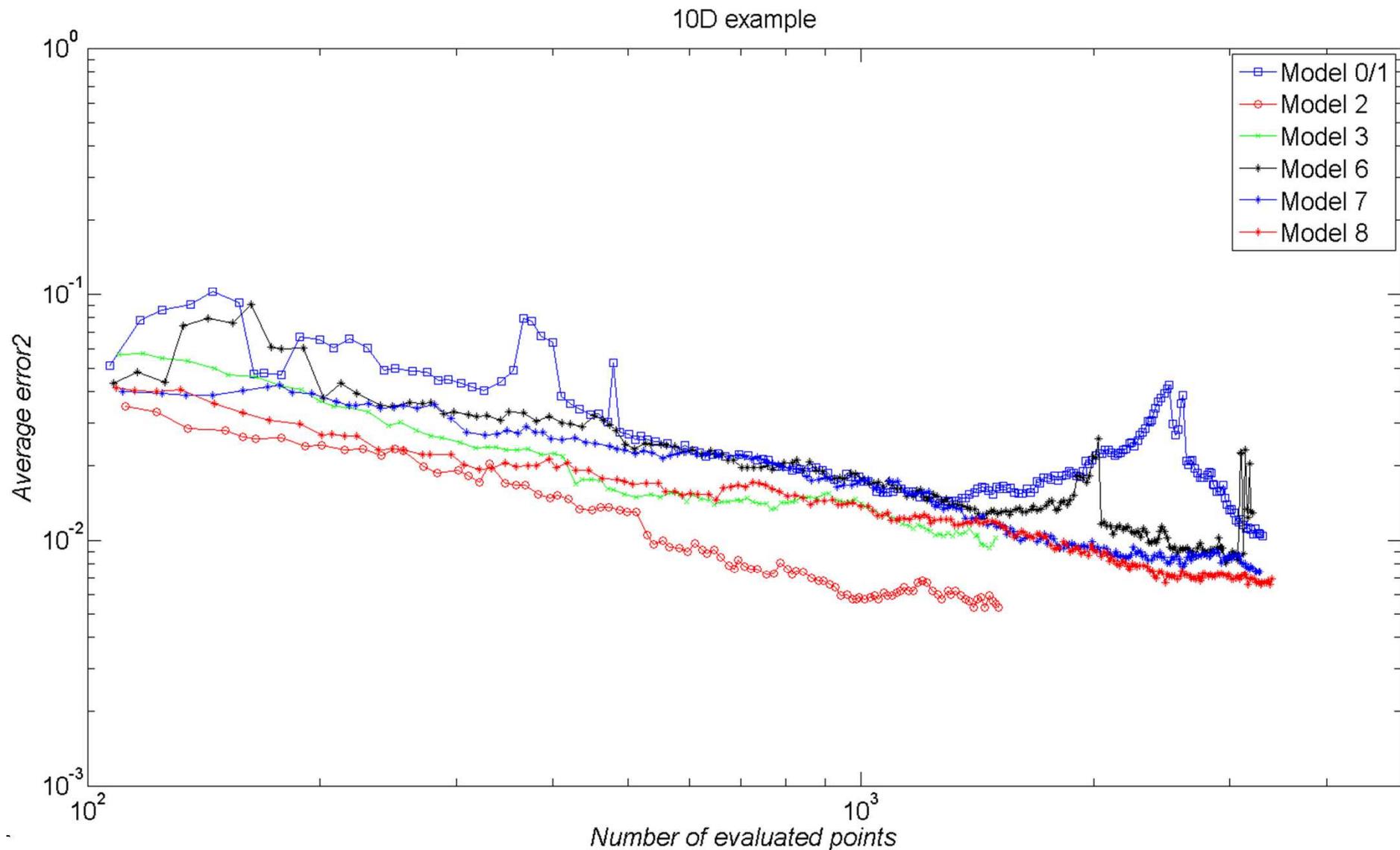
RBFN (Matlab)



Kriging



Quality of updating procedure



Reference

- [1] Jin, Y. (2003) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing Journal*, 9(1):3-12.
- [2] T. W. Simpson, J. D. Peplinski, P. N. Koch and J. K. Allen. (2001) Metamodels for Computer-based Engineering Design: Survey and recommendations. *Engineering with Computers* 17: 129–150.
- [3] H. Nakayama, K. Inoue and Y. Yoshimori (2004) Approximate optimization using computational intelligence and its application to reinforcement of cable-stayed bridges, ECCOMAS.
- [4] M. K. Karakasis and K. C. Giannakoglou (2004) On the use of surrogate evaluation models in multi-objective evolutionary algorithms, ECCOMAS.
- [5] Ibrahimbegović, A., Knopf-Lenoir, C., Kučerová, A., Villon, P., (2004) Optimal design and optimal control of elastic structures undergoing finite rotations, *International Journal for Numerical Methods in Engineering*.

Reference

- [6] Forrester, A., Sobester A., and Keane A., (2008) Engineering design via surrogate modelling: a practical guide. John Wiley & Sons.
- [7] Weise, Thomas, et al. "Why is optimization difficult?" *Nature-Inspired Algorithms for Optimisation*. Springer Berlin Heidelberg, 2009. 1-50.
- [8] D. C. Montgomery. Design and Analysis of Experiments, 5th Edition. Wiley, June 2000.
- [9] R. H. Myers and D. C. Montgomery. Response Surface Methodology: Process and Product Optimization Using Designed Experiments. New York: Wiley, 1995.

Při přípravě této přednášky byla použita řada materiálů laskavě poskytnutých Ing. Adélou Pospíšilovou ze Stavební fakulty ČVUT. **Ostatní zdroje jsou ocitovány v místě použití.**

Prosba. V případě, že v textu objevíte nějakou chybu nebo budete mít námět na jeho vylepšení, ozvěte se prosím na **`matej.leps@fsv.cvut.cz`**.

Datum poslední revize: 18.12.2019

Verze: 001