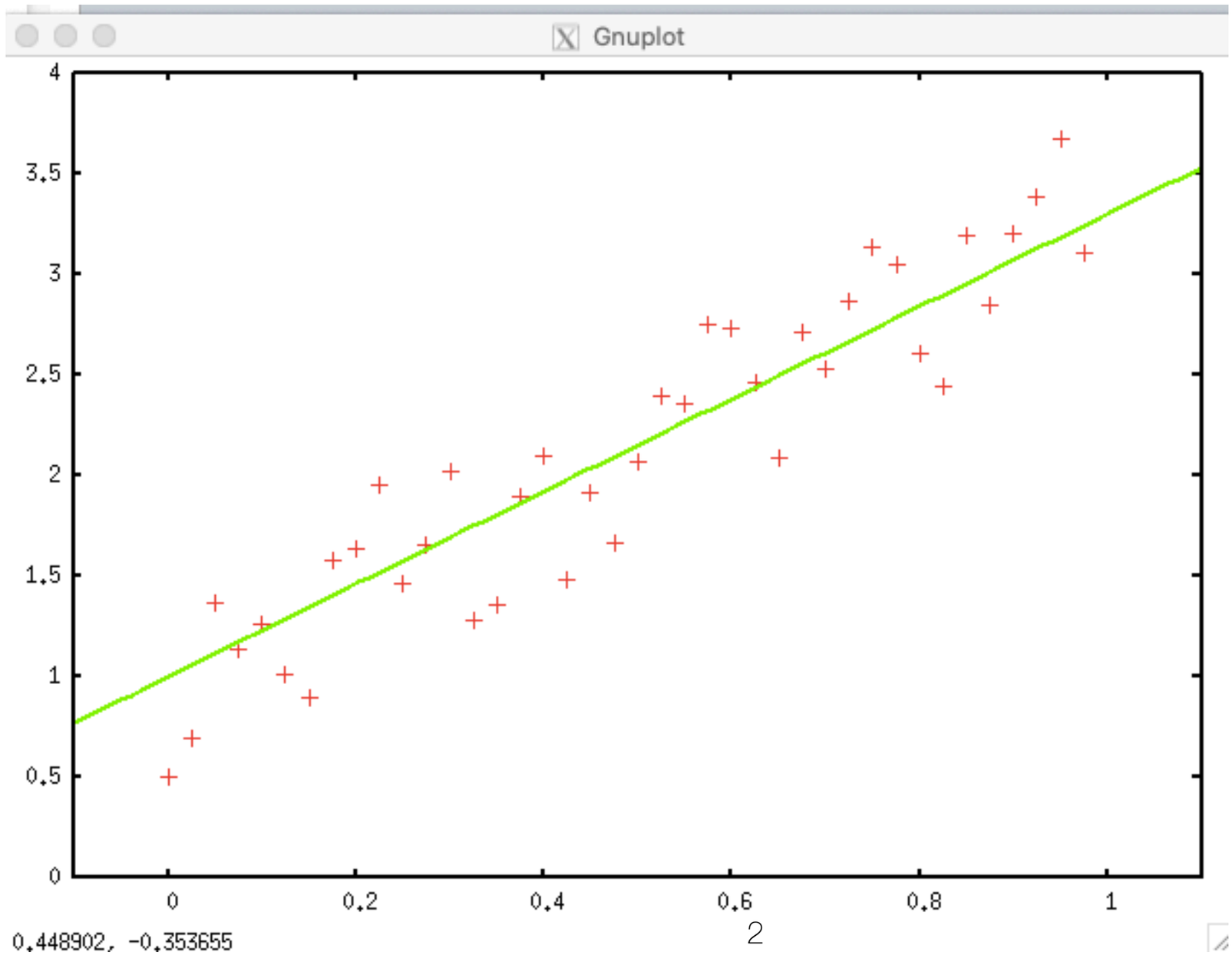
Fitovanie dát, minimalizácia funkcie, numerické chyby

Jaroslav Tóbik
Elektrotechnický ústav SAV
jaroslav.tobik@savba.sk

Fitovanie dát

sada dát (spolu s odhadom presnosti (neurčitosti)) $\{(x_i, y_i, \sigma_{y_i})\}_{i=1}^{N}$

fitovanie lineárnou funkciou typu: $y = ax + b$



Metóda najmenších štvorcov

Penalty function (cost f.) $\chi(a,b) = \sum_{i=1}^{N}(y_i - ax_i - b)^2$ $\{(x_i, y_i, \sigma_{y_i})\}_{i=1}^{N}$

$\{y_i\}$ sada nameraných dát $ax_i + b = y(x_i)$ fitovacia funkcia (a,b) fitovacie parametre

$$\frac{\partial\chi(a,b)}{\partial a} = -2\sum_{i=1}^{N}(y_i - ax_i - b)x_i = 0 \qquad a\sum_{i=1}^{N}x_i^2 + b\sum_{i=1}^{N}x_i = \sum_{i=1}^{N}x_i y_i$$

$$\frac{\partial\chi(a,b)}{\partial b} = -2\sum_{i=1}^{N}(y_i - ax_i - b) = 0 \qquad a\sum_{i=1}^{N}x_i + b\sum_{i=1}^{N}1 = \sum_{i=1}^{N}y_i$$

$$S_x = \sum_{i=1}^{N}x_i \quad S_{xx} = \sum_{i=1}^{N}x_i^2 \quad S_{xy} = \sum_{i=1}^{N}x_i y_i \quad N = \sum_{i=1}^{N}1$$

$$aS_{xx} + bS_x = S_{xy}$$
$$aS_x + bN = S_y$$

$$a = \frac{S_{xy}N - S_x S_x}{S_{xx}N - S_x S_x}$$
$$b = \frac{S_{xx}S_x - S_{xy}S_x}{S_{xx}N - S_x S_x}$$

Zovšeobecnenie - stále lineárne fitovanie

Sada nameraných dát $\{(x_i, y_i, \sigma_{y_i})\}_{i=1}^{N}$

Sada bázových funkcií $\{f_j\}_{j=1}^{M}$

Fitovacia funkcia: $f(y_i) = \sum_{j=1}^{M} a_j f_j(x_i)$

Cost function: $\chi(\mathbf{a}) = \sum_{i=1}^{N} (f(x_i, \mathbf{a}) - y_i)^2$

Nutná podmienka extrému: $\frac{\partial \chi}{\partial a_j} = 0$

$$
\begin{aligned}
a_1 S_{f_1 f_1} + a_2 S_{f_1 f_2} + \cdots + a_M S_{f_1 f_M} &= S_{f_1 y} \\
a_1 S_{f_1 f_2} + a_2 S_{f_2 f_2} + \cdots + a_M S_{f_2 f_M} &= S_{f_2 y} \\
&\vdots \\
a_1 S_{f_1 f_M} + a_2 S_{f_2 f_M} + \cdots + a_M S_{f_M f_M} &= S_{f_M y}
\end{aligned}
$$

Čím viac fitovacích funkcií - tým viac sa fit blíži k experimentálnym dátam, ale neistoty koeficientov vo všeobecnosti rastú!

Odhad neistoty fitovacích parametrov

$$\chi(a,b) = \sum_{i=1}^{N} \frac{(y_i - y(x_i))^2}{\sigma_i^2}$$

Ako veľmi sa mení fitovací parameter ak mením “namerané” hodnoty?

$$\sigma_a^2 = \sum_{i=1}^{N} \left(\frac{\partial a}{\partial y_i} \right)^2 \sigma_{y_i}^2$$

$$\sigma_b^2 = \sum_{i=1}^{N} \left(\frac{\partial b}{\partial y_i} \right)^2 \sigma_{y_i}^2$$

$$a = \frac{S_{xy} N - S_x S_x}{S_{xx} N - S_x S_x}$$

$$b = \frac{S_{xx} S_x - S_{xy} S_x}{S_{xx} N - S_x S_x}$$

$$\frac{\partial a}{\partial y_i} = \frac{x_i N}{S_{xx} N - S_x S_x}$$

$$\frac{\partial b}{\partial y_i} = \frac{-x_i S_x}{S_{xx} N - S_x S_x}$$

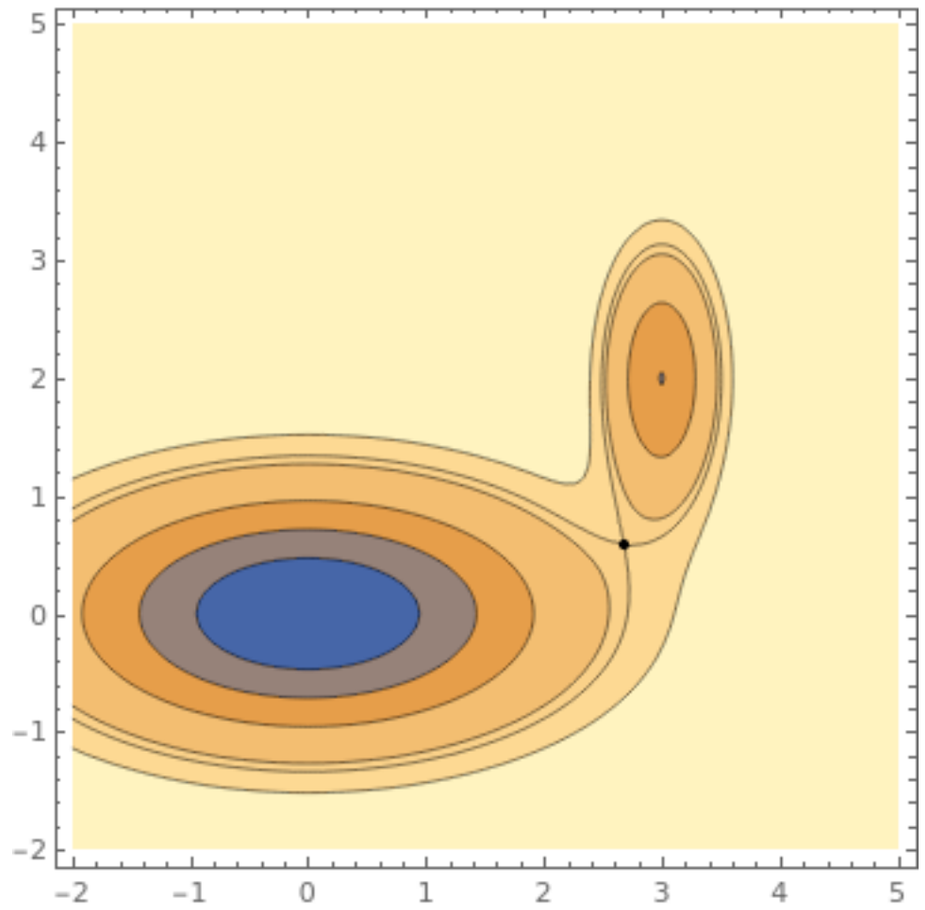
$$S_x = \sum_{i=1}^{N} x_i \quad S_{xx} = \sum_{i=1}^{N} x_i^2 \quad S_{xy} = \sum_{i=1}^{N} x_i y_i \quad N = \sum_{i=1}^{N} 1$$

Lineárne fitovanie je štandardne súčasťou tabuľkových procesorov (funkcia LINEST v Micro$oft Exceli) - len keby ste chceli robiť neštandarné fitovanie, tak budete vedieť ako a prečo.

Nelineárne fitovanie

Penalty function závisí obecne nelineárne od viacerých parametrov - obecný problém hľadania minima funkcie v mnohodimenziálnom priestore (parametrov).

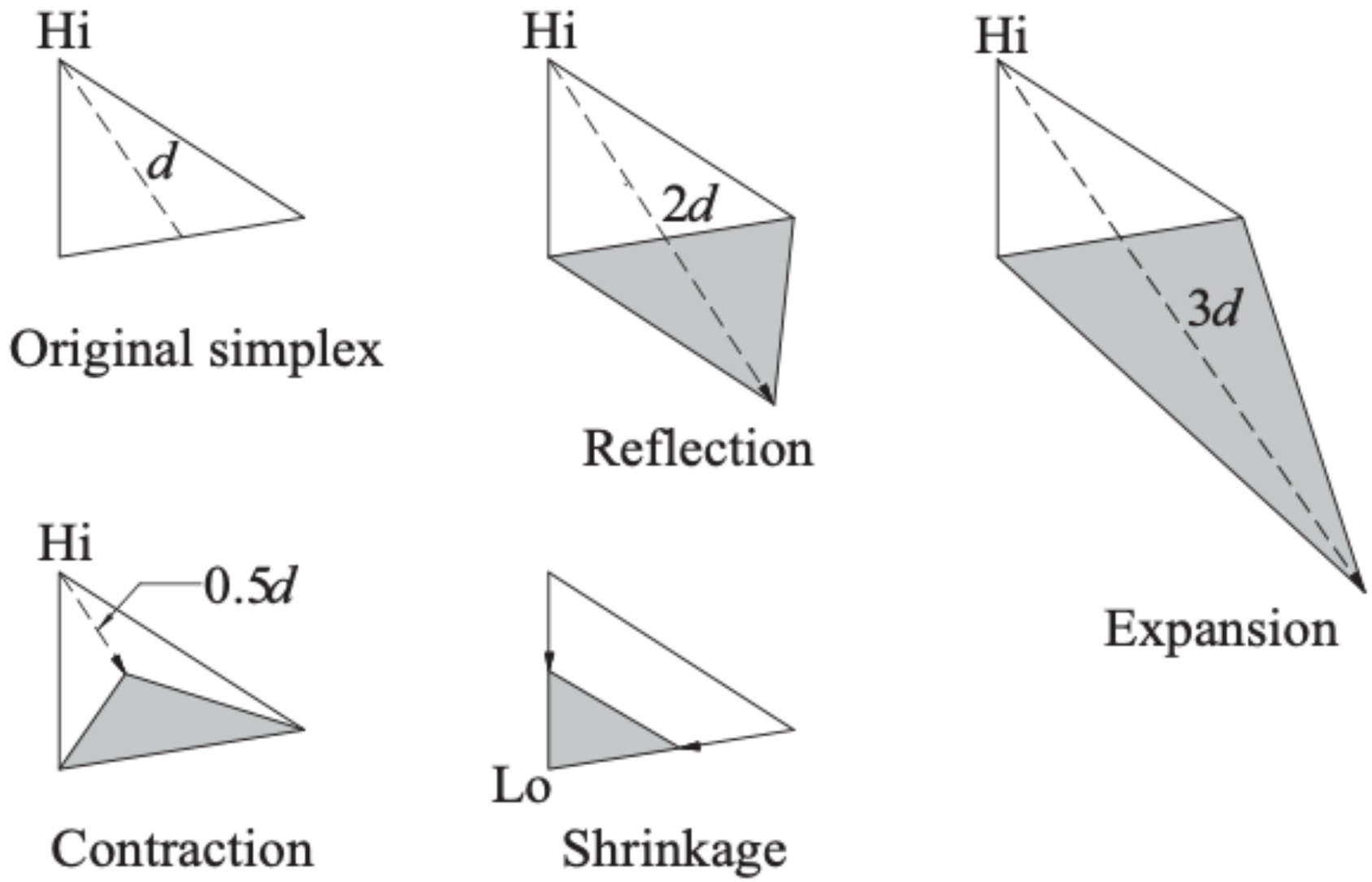
1.problém: miním môže byť viac ako jedno
2.problém: dimenzionalita rýchlo zväčšuje objem fázového priestoru



Metódy nahľadanie minima

1. nepotrebujú derivácie funkcie podľa parametra
2. potrebujú derivácie funkcie podľa parametra
3. stochastické metódy čo vedia výjsť z lokálneho minima a prezrieť priestor parametrov viac “globálne”

Downhill Simplex (Nelder–Mead) Method



Downhill Simplex (Nelder–Mead) Method

1. Pre M dimenzionálny priestor štartujeme s M+1 bodmi, body x zoradíme podľa veľkosti

   $\{x_1, x_2, \ldots, x_{M+1}\} \quad f(x_1) \leq f(x_2) \leq \ldots f(x_{M+1})$

2. Z bodov x_1 až x_M vypočítame ťažisko: $x_0 = \frac{1}{M} \sum_{i=1}^{M} x_i$

3. Reflexia: bod x_{M+1} odzrkadlíme cez ťažisko: $x_r = x_0 - (x_{M+1} - x_0)$

   Ak $f(x_1) \leq f(x_r) < f(x_N)$ zoberieme x_r do novej sady bodov a ideme na bod 1

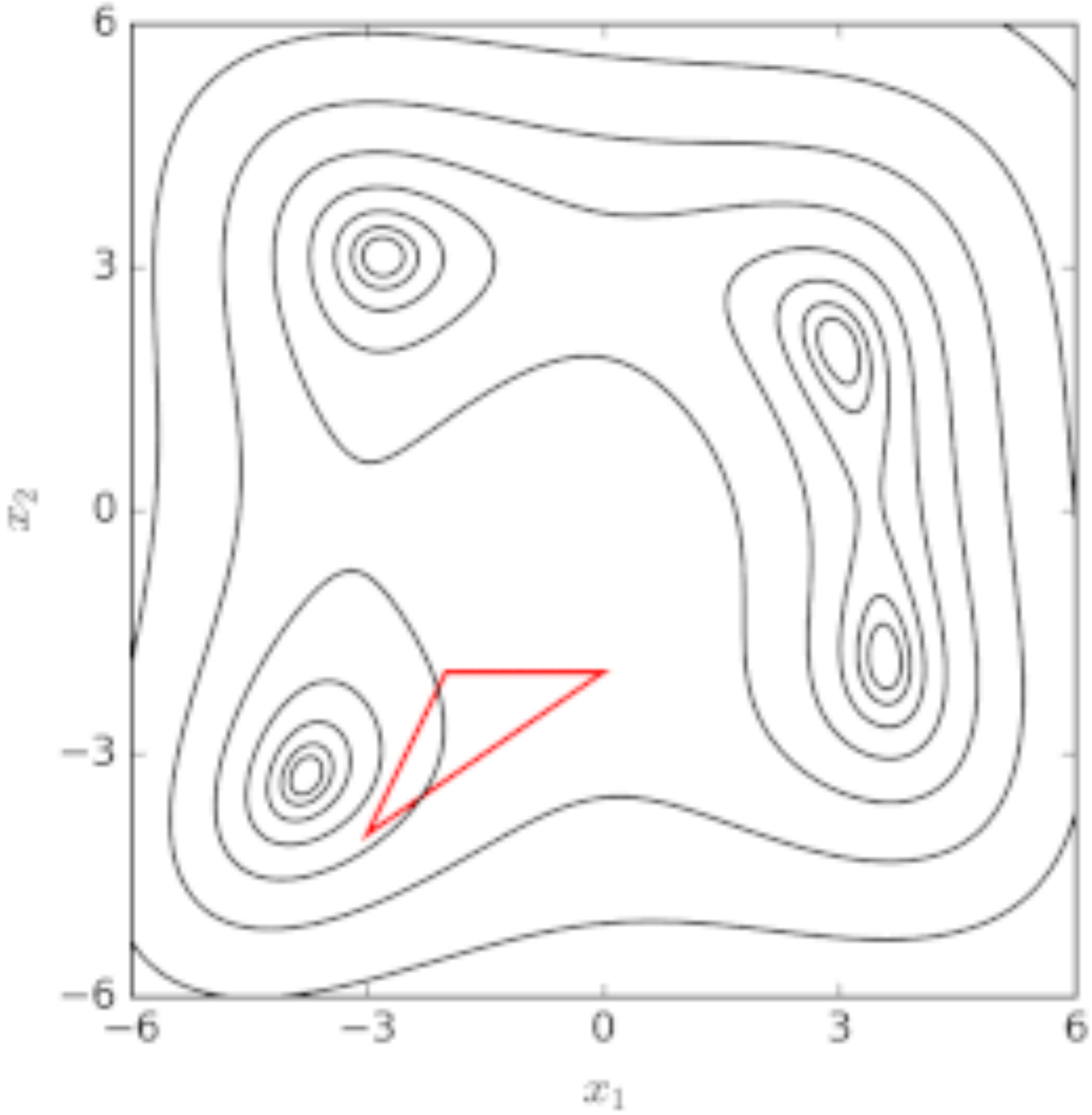
4. Expanzia: ak $f(x_r) < f(x_1)$ $x_e = x_0 - 2(x_{M+1} - x_0)$

   Ak $f(x_e) < f(x_r)$ prijmeme bod x_e do sady $\{x_1, \ldots, x_{M+1}\}$ s tým, že vyhodíme x_{M+1}

   Ak $f(x_e) \geq f(x_r)$ prijmeme bod x_r do sady (reflexia) a ideme na krok 1

5. Kontrakcia: Ak $f(x_r) < f(x_{M+1})$ $x_c = x_0 - 0.5(x_{M+1} - x_0)$
   Ak $f(x_c) < f(x_r)$ prijmeme x_c do sady a ideme na krok 1

6. Preškálovanie: vytvoríme novú sadu $x'_i = x_1 + 0.5(x_i - x_1)$ a ideme na krok 1



https://en.wikipedia.org/wiki/Nelder–Mead_method

Downhill Simplex (Nelder–Mead) Method

Výhody

- ešte stále pomerne jednoduchý algoritmus
- robustná metóda, pomerne jednoducho odladiteľná
- nepotrebuje gradienty

Nevýhody

- pomalšia konvergencia k minimu - neoplatí sa ak počítanie f(x) je náročné
- efektivita prudko klesá s počtom dimenzií (ťaháte k minimu celý simplex - gradientné metódy idú len s jedným bodom)

Stochastická metóda - Simmulated Anealing

Metropolis Monte-Carlo algoritmus

$$x_i \rightarrow x_{i+1} = x_i + \delta$$

$$P(x_i \rightarrow x_{i+1}) = 1 \text{ if } f(x_{i+1}) \leq f(x_i)$$

$$P(x_i \rightarrow x_{i+1}) = \exp\left(-\frac{f(x_{i+1}) - f(x_i)}{kT}\right) \text{ otherwise}$$

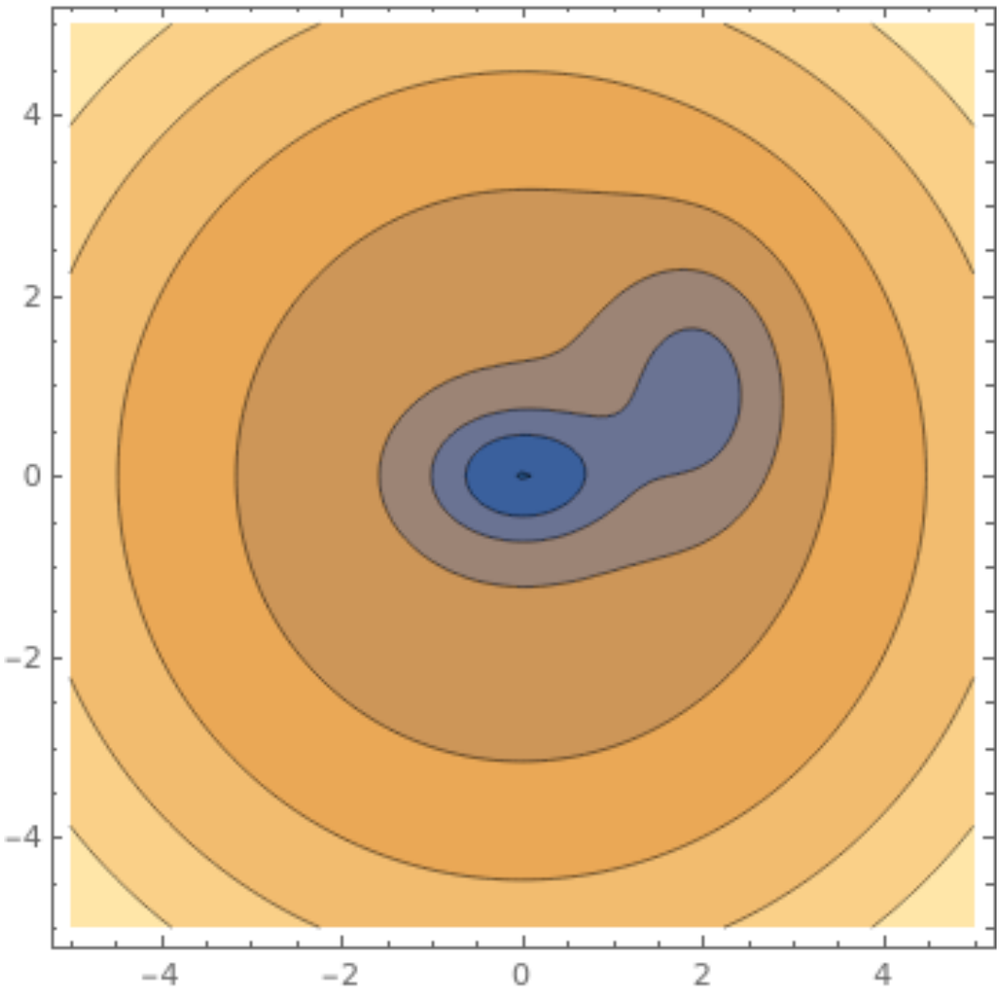
Simmulated annealing T znizujeme z vysokej hodnoty na nízku

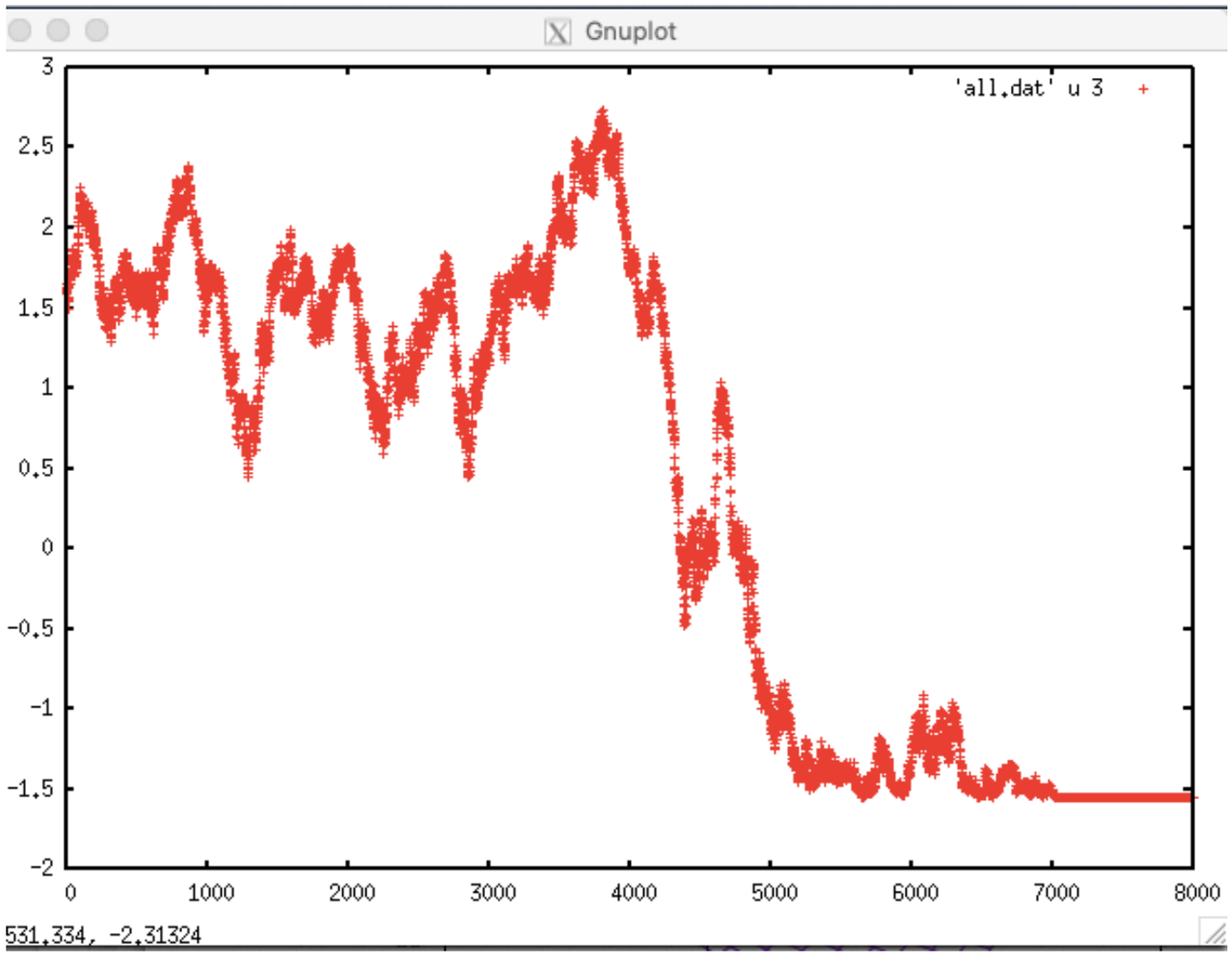
Výhody

- extrémne jednoduchý algoritmus
- robustná metóda
- nepotrebuje gradienty
- funguje pri mnoho-dimenzionálnych a diskrétnych problémoch
- dokáže “vyliezť” z lokálneho minima a hľadať iné - lepšie (snáď)

Nevýhody

- veľmi pomalá konvergencia (veľmi veľa výpočtov f(x))
- potrebuje generovať náhodné čísla s dobrou kvalitou náhodnosti





Konvergencia algoritmu simulované žíhanie. Vľavo je mapa funkcie ktorej minimum sme hľadali, vpravo je vývoj energie ako funkcie čísla iterácie. Po 700 krokoch bola teplota T=0 a algoritmus už musel konvergovať iba k najbližšiemu minimu.

Numerická presnosť

single precision (float v jazyku C, REAL vo Fortrane) má takmer 7 platných číslic
double precision (double v C aj vo Fortrane) má 15
zdá sa to dosť ale…

$$\int_0^{L_x}\int_0^{L_y}\int_0^{L_z} f(x,y,z)dxdydz \approx \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{N_z} f(x_i,y_j,z_k)\frac{L_xL_yL_y}{N_xN_yN_z}$$

sčitávame čísla rôznych rádov - z menšieho čísla sa desatinné miesta urežú! (tzv. round-off error, zaokrúhľovacia chyba) a ak sa kumuluje - zmenšenie kroku môže viesť z zhoršeniu výsledku - treba odhadnúť kam môžeme ísť s presnosťou (slušná literatúra upozorní na limity danej metódy).

$S + \Delta \quad \frac{S}{\Delta} \approx N_xN_yN_z$ **už pre Nx=Ny=Nz=100 mi single precision nestačí!!!**

Tolerancia na určenie polohy minima - len polovica platných desatinných miest!

$$f(x) \approx f(x_0) + \frac{1}{2}f''(x_0)(x-x_0)^2$$

Ďakujem za pozornosť!

Literatúra:

W.H. Press et.al.: Numerical Recipes in Fortran (C, Python, ???)