Iskanje ekstremov funkcij in prilagajanje funkcij meritvam

9. 3. 2022

4. naloga: Iskanje funkcijskih ekstremov in prilagajanje funkcij meritvam

Tako kot pri vseh numeričnih metodah kompleksnost metode iskanja ekstremov narašča z dimenzijo prostora v katerem je funkcija definirana. Najbolj robustne in zanesljive metode so tako razvite za iskanje ekstremov v eni dimenziji. V splošnem pa ne smemo pozabiti na dejstvo, da *ne obstaja* metoda, ki bi avtomatsko našla *globalni* ekstrem (minimum ali maksimum), vsaka metoda se bo slej ko prej ustavila v nekem lokalnem ekstremu. Ali je le-ta globalen in/ali edini pa je potrebno proučiti naknadno (s poskušanjem ipd.).

Iskanje minimumov ali maksimumov funkcije lahko poenotimo z metodami iskanja minimumov, saj za maksimume lahko vedno definiramo $f \to -f$.

V eni dimenziji je najpreprostejša (in najrobustnejša) metoda *zlatega reza*. Najprej poiščemo tri točke (A<b<C), za katere velja f(b) $<$ f(A) in f(b) $<$ f(C); točka b je torej groba ocena minimuma. Novo točko B nato najdemo s uporabo razmerja zlatega reza, tukaj podano kot:

$$\frac{1}{\varphi} = \frac{\sqrt{5}-1}{2} = 0.61803.$$

Koordinata nove točke B je podana z:

$$B = A + h \cdot \frac{1}{\varphi}, \;\; h = |C - A|.$$

Dodatno poskusno točko D pa nato dodamo s faktorjem:

$$\frac{1}{\varphi^2} = \frac{3-\sqrt{5}}{2} = 0.38197,$$

in sicer s koordinato:

$$D = A + h \cdot \frac{1}{\varphi^2}, \;\; h = |C - A|.$$

Očitno je točka D bližje točki A kot točka B. Nato obravnavamo dva scenarija:

- Če je vrednost $f(D) > f(B)$, potem postane novi triplet (D,B,C), točko A torej nadomesti točka D.
- V nasprotnem primeru, ko je $f(D) < f(B)$, pa postane novi triplet (A,D,B), točko C torej nadomesti točka B, njeno vlogo pa prevzame točka D.

Pri vsakem koraku se interval zmanjša za faktor $\frac{1}{\varphi}$.

Metodo nato ponavljamo do želene natančnosti, pri čemer moramo paziti, da je dosegljiva natančnost le *koren iz numerične natančnosti računskih operacij*! (npr 10^{-8} za dvojno natančnost (double precision)).

Hitrejša, a nekoliko labilnejša metoda, je iskanje minimumov z prilagajanjem parabole na tri začetne točke in oceno novega minimuma iz minimuma prilagojene parabole:

$$x = b - \frac{1}{2}\frac{(b-a)^2[f(b)-f(c)] - (b-c)^2[f(b)-f(a)]}{(b-a)[f(b)-f(c)] - (b-c)[f(b)-f(a)]},$$

ki pa odpove(duje) ko so izbrane tri točke (skoraj) kolinearne. Smiselna je torej ustrezna kombinacija obeh metod, poznana kot *Brentova metoda*.

V primeru, ko je poznana analitična oblika odvoda iskane funkcije, si seveda lahko pomagamo z numeričnim iskanjem ničle v odvodu (npr z metodo bisekcije).

V več dimenzijah je problem ustrezno težji, najrobustnejša in precej učinkovita je t.i. *Downhill simplex* metoda, ki jo bralec z lahkoto najde v literaturi in na spletu.

Poseben primer uporabe minimizacije funkcij je prilagajanje (angl. *fitting*) modelov (in s tem napovedanih funkcij) s prostimi parametri naboru izmerjenih količin. Najpogostejša načina prilagajanja funkcij s prostimi parametri podatkom sta metodi χ^2 in metoda največje zanesljivosti (angl. *Maximum Likelihood Metod*). Označimo nabor merjenih podatkov z $\{x_i, y_i\}$ in njihove nedoločenosti z $\{\sigma_i\}$, modelske funkcije s $y = f(x, \alpha)$, kjer so z α označeni neznani parametri, ter uvedimo cenilko (angl. *estimator*) za χ^2:

$$\chi^2(\alpha) = \sum_i \frac{(y_i - f(x_i, \alpha))^2}{\sigma_i^2},$$

kjer je nato ocena iskanih parametrov α dobljena z minimizacijo funkcije $\chi^2(\alpha)$.
Metoda največje zanesljivosti se uporablja predvsem v primerih, ko vrednost $y = f(x, \alpha)$ predstavlja verjetnost (ali verjetnostno gostoto) in tako iščemo nabor parametrov α, pri katerih je nabor verjetnosti za dane izmerke največji. Iščemo torej maksimum produkta verjetnosti

$$L(\alpha) = \prod_i f(x_i, \sigma_i, \alpha),$$

oziroma še bolj pogosto minimum naravnega logaritma zgornjega izraza (ki je pohlevnejša funkcija):

$$-\ln L(\alpha) = -\sum_i \ln f(x_i, \sigma_i, \alpha).$$

V primeru, da je verjetnostna porazdelitev Gaussova in da iščemo neznani parameter le-te (μ, srednja vrednost), sta obe metodi enaki (dokaz je prepuščen bralcu).

Naloga:

- Z metodo zlatega reza in parabolično ter saj eno vgrajeno metodo (npr Brentovo metodo v SciPy: scipy.optimize.brent) poišči ekstreme nekaj funkcij različnih redov tipa $x^n * sin(x)$ (recimo do tretjega reda) in primerjaj hitrost (št. korakov) in natančnost s točnimi vrednostmi.

- Z zgornjimi metodami poišči najboljše prilagajanje med podatki:

$$\{x_i\} = [0.1, 0.41052632, 0.72105263, 1.03157895, 1.34210526, 1.65263158,\\ 1.96315789, 2.27368421, 2.58421053, 2.89473684, 3.20526316, 3.51578947,\\ 3.82631579, 4.13684211, 4.44736842, 4.75789474, 5.06842105, 5.37894737,\\ 5.68947368, 6.]$$

$$\{y_i\} = [0.04959185, 0.15562821, 0.22546384, 0.29475305, 0.3564775, 0.40037302, 0.45322831, 0.44861781, 0.47145439, 0.49542899, 0.48391543, 0.50759507, 0.49552151, 0.50198301, 0.44788353, 0.46090299, 0.43101869, 0.44916611, 0.40895603, 0.39695574]$$

$$\{\sigma_i\} = [0.03682367, 0.0014194, 0.01994378, 0.01061233, 0.0100865, 0.00231544, 0.01935984, 0.00635483, 0.02376167, 0.00087145, 0.02262165, 0.00350645, 0.01716223, 0.02109906, 0.02477181, 0.03953272, 0.02218641, 0.01454135, 0.01634465, 0.04684965]$$

in funkcijo dveh neznanih parametrov a in b, oblike $f(x, a, b) = a*x*exp(b*x)$. Uporabi eno od vgrajenih metod (npr. v Pythonovih SciPy knjižnicah.)

- **Dodatno:** Z Downhill simplex metodo poišči maksimum dvo-dimenzionalne Gaussove porazdelitve ter dvojne 2D Gaussove porazdelitve (kamelja grba).

Iskanje ekstremov funkcij (v 1D)

Naloga: Imamo funkcijo $f(\vec{x})$, ki je funkcija n neodvisnih spremenljivk $\vec{x} = (x_1, x_2, x_3, \ldots x_n)$. Določiti želimo vrednosti $x_1, x_2, \ldots x_n$, kjer funkcija doseže maksimum ali minimum.

Ekstreme funkcije (maksimume, minimume) delimo na:

- GLOBALNE: dejansko največja oziroma najmanjša vrednost funkcije na njenem definicijskem območju.

- LOKALNE: največja oziroma najmanjša vrednost funkcije v neki kameni okolici / znotraj kakšnega intervala, izvzemši robove (robne točke) opazovanega intervala.

Mi se bomo spoznali z metodami za iskanje LOKALNIH EKSTREMOV!

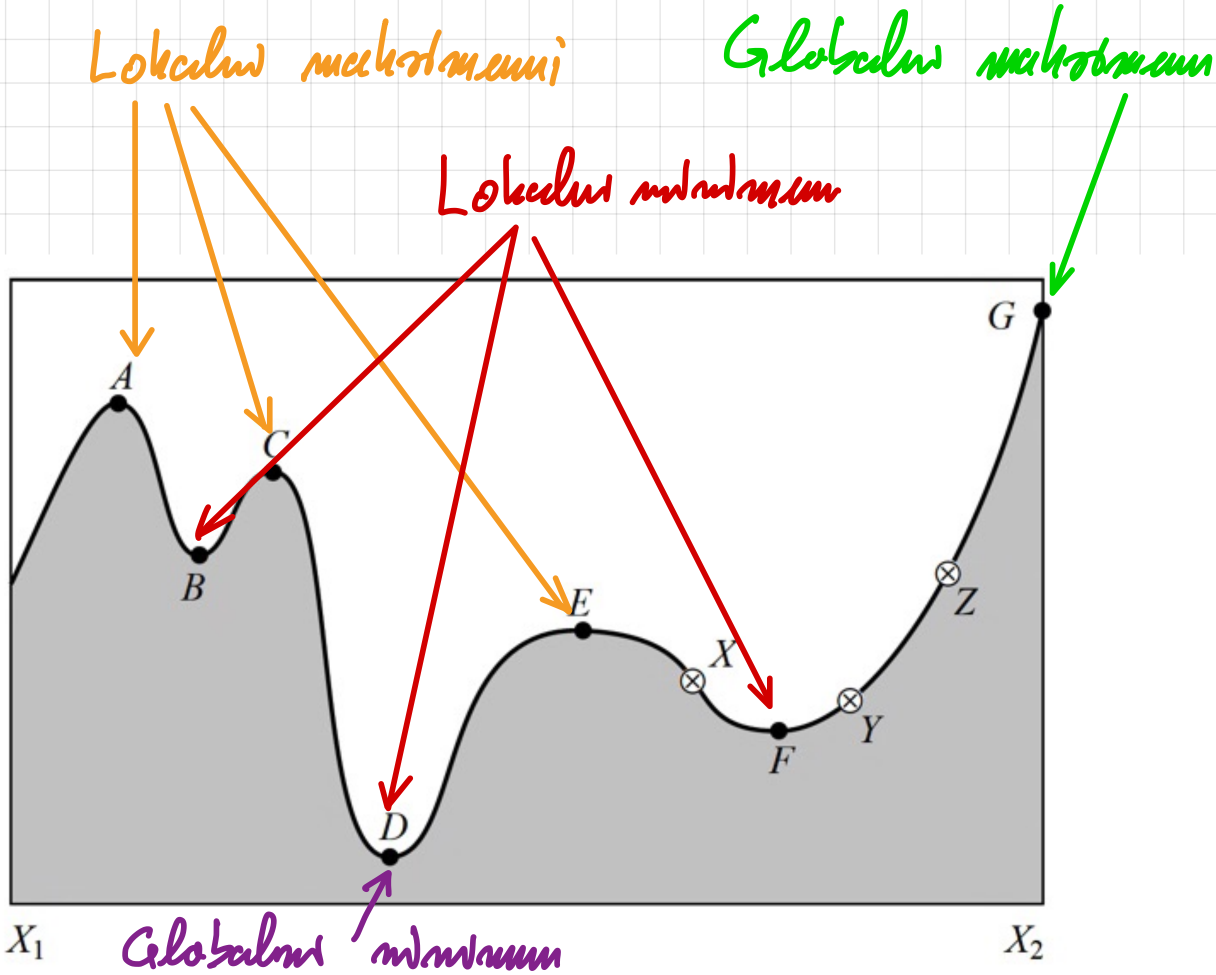


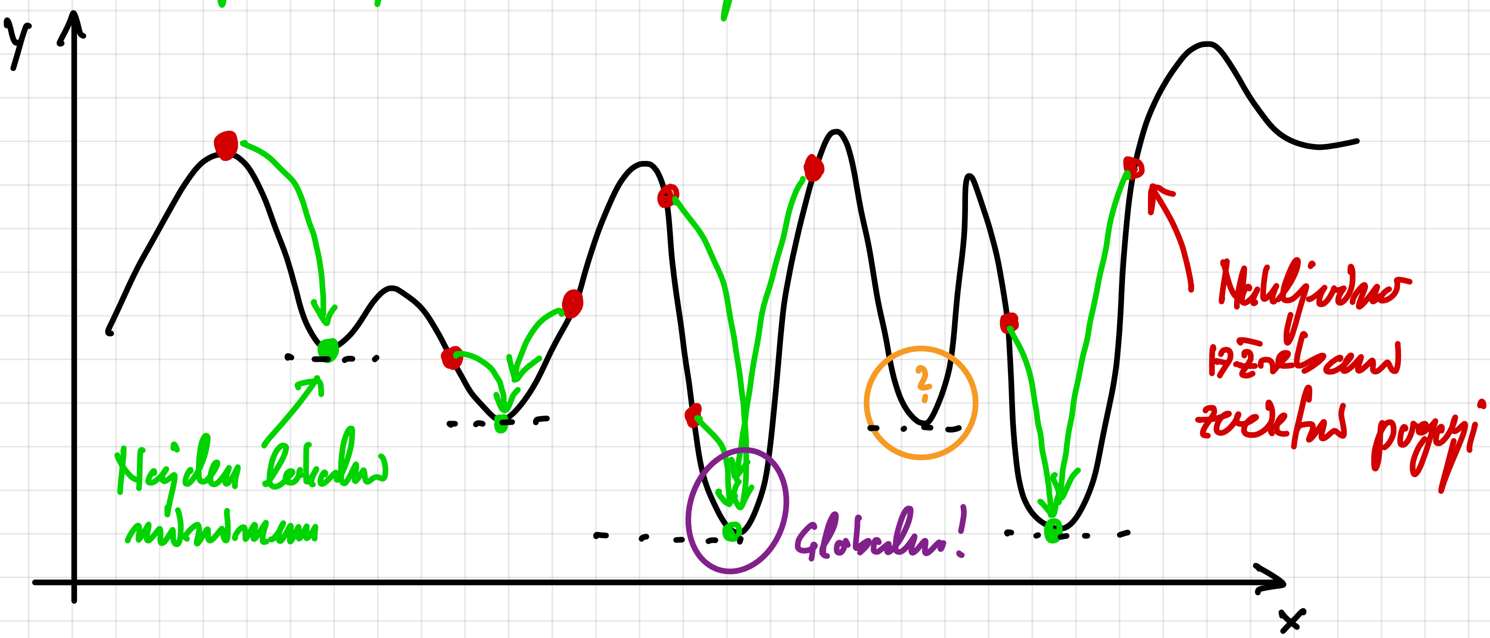
Figure 10.0.1. Extrema of a function in an interval. Points A, C, and E are local, but not global maxima. Points B and F are local, but not global minima. The global maximum occurs at G, which is on the boundary of the interval so that the derivative of the function need not vanish there. The global minimum is at D. At point E, derivatives higher than the first vanish, a situation which can cause difficulty for some algorithms. The points X, Y, and Z are said to "bracket" the minimum F, since Y is less than both X and Z.

Iskanje globalnih ekstremov (predvsem v večjih dimenzijah) je zelo zahtevno. Težave so:

- da se zataknemo v lokalnem ekstremu, ne najdemo poti iz njega in ga proglasimo za globalnega.
- med iskanjem globalnega ekstrema, najdemo vrsto lokalnih, globalnega pa spregledamo.

Predlagana taktika za iskanje globalnega ekstrema

1.) Poiščemo lokalne ekstreme za množico zelo različnih začetnih vrednosti neodvisnih spremenljivk in izberemo tistega, v katerem je vrednost funkcije najmanjša! Začetne vrednosti lahko žrebamo kar naključno po definicijskem območju!

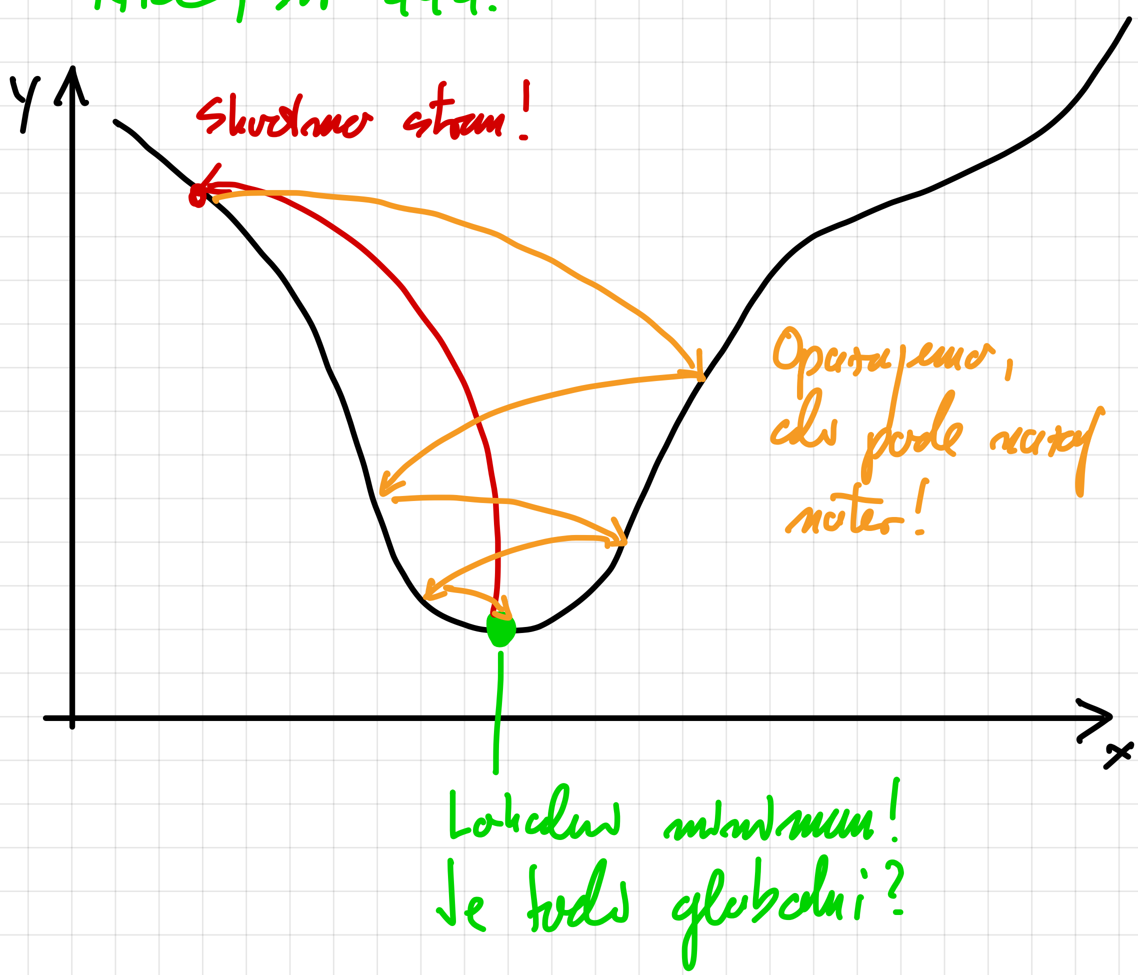


Pregledamo vse lokalne minimume in izberemo tistega, ki je najnižji:

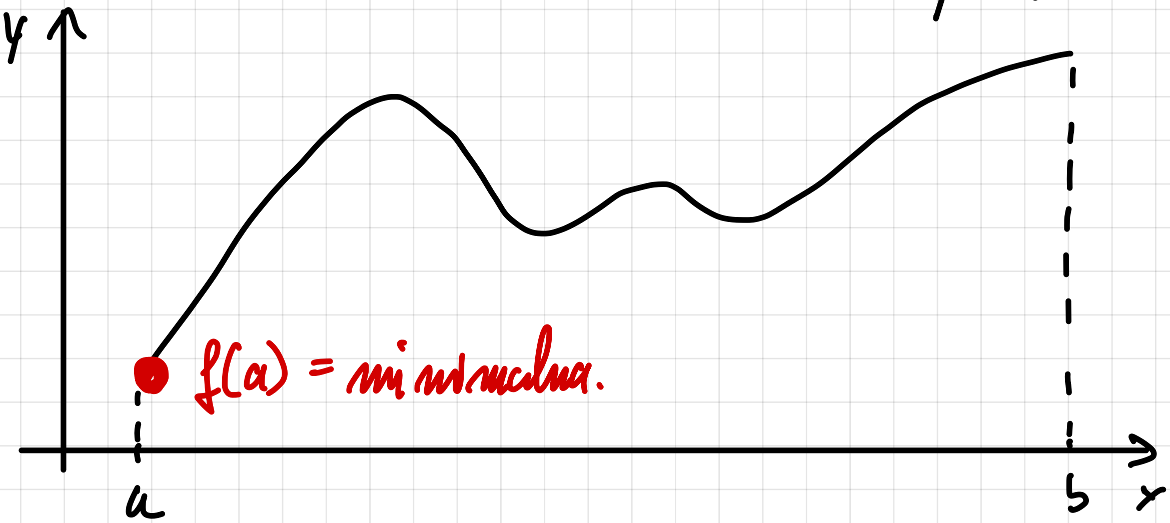
- Lahko se zgodi, da kakšen minimum spregledamo! Zato moramo analizo izvesti za dovolj različnih začetnih pogojev (ali pa za dovolj gosto mrežo točk) kar še vedno ni zagotovilo, da česa ne izpustimo!

– Lahko se zgodi, da je več lokalnih minimumov enako globalnih! ⇒ Povdamo vse!

2.) Izvedemo perturbacijo lokalnega ekstrema: Za neko random amplitudo se odmaknemo od ekstrema (npr. minimuma) in pogledamo ali metoda konvergira k boljši točki ali se vedno vrne k isti (izbranemu lokalnemu ekstremu) ⇒ Po tem principu delujejo metode, ki delujejo po principu "simuliranega ohlajanja". Za več glej NRC, str 444.



3.) Ne pozabi preveriti vrednosti funkcije na robu intervala (definicijskega območja)



Globalni ekstrem se lahko skriva na robu intervala (npr. [a, b] pri f(a)).

<u>Opomba</u>: Iskanje minimumov je tesno povezano z iskanjem maksimumov: Iskanje minimuma funkcije f je enako iskanju maksimuma funkcije $-f$.

Algoritmi za računanje ekstremov so večinoma napisani tako, da iščejo minimume. Maksimume dobimo z iskanjem minimuma $(-f)$!

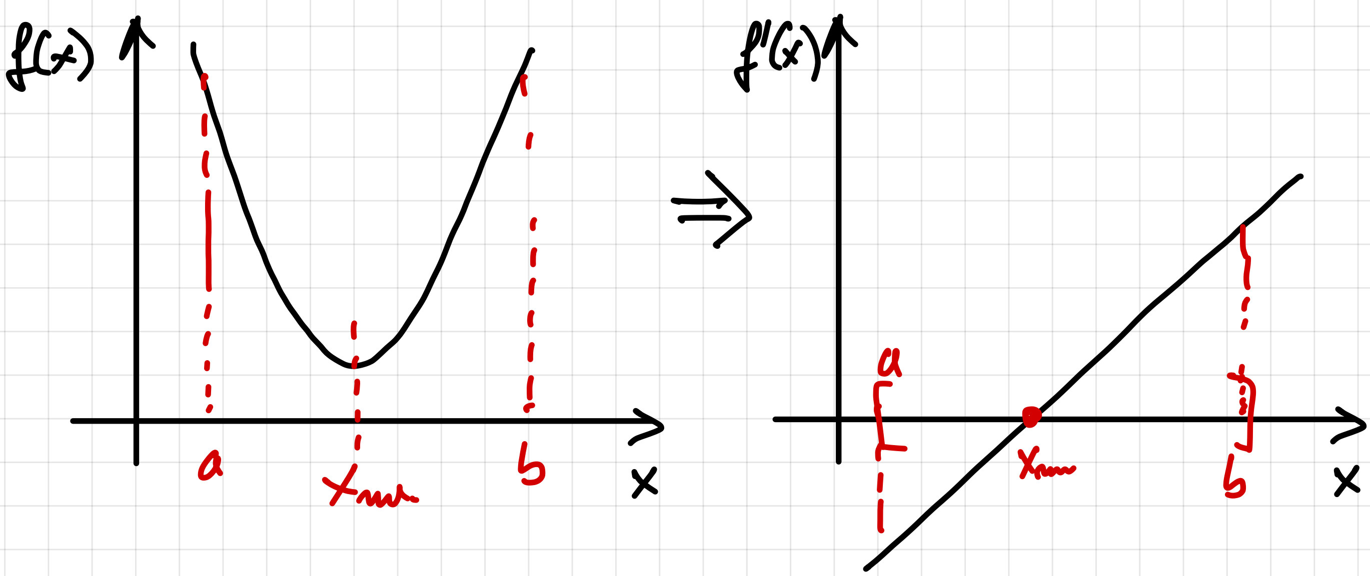
Algoritmi za iskanje minimumov v 1D:

- Algoritme ločimo na tiste, ki za določanje minimuma potrebujejo le funkcijske vrednosti $f(x_i)$ i $\{x_i\}$ in tiste, ki poleg funkcijskih vrednosti uporabljajo tudi odvode $f'(x_i)$
  Če jih znamo izvrednotiti!?

- Metode, ki upoštevajo odvode spadajo med močnejša orodja vendar:
  - V praksi se jih ne splača uporabljati, saj dodatno računanje odvodov, podaljša konvergenco do te mere, da so metode, ki upoštevajo le funkcijske vrednosti hitrejše
  - Odvode se ne uporablja za določevanje parametrov polinomov višjih redov (skupaj z vrednostmi funkcij), s katerimi bi radi bolje modelirali potek funkcije in s tem zagotovili hitrejšo konvergenco.
    Polinomi višjih redov se lahko divje obnašajo, in nas vržejo izven iskalnega območja.

– odvode se uporablja za hitrejše iskanje naslednjega približka minimum v iterativnem procesu. Uporaba odvoda je omejena! (Glej NRC, poglavje 10.3, str. 405)

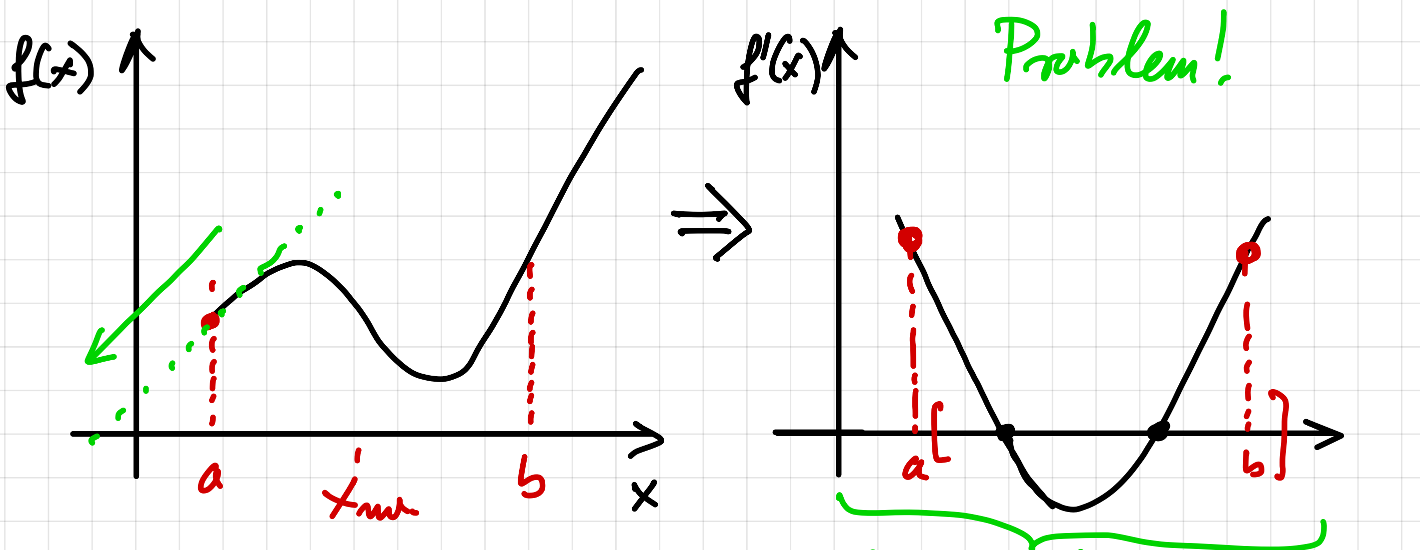
Pozor: Če poznamo odvode, potem bi se dalo minimum iskati tudi z iskanjem ničel odvoda z uporabo zmogljivih metod za iskanje ničel.



Pri taki metodi nas čakajo pasti:

- Če iščemo $f'(x)=0$, potem a priori ne vemo, ali smo našli maksimum, ali minimum?

- Kako napredujemo od začetnih pogojev če ena ali več začetnih točk kaže, da je smer napačna v smer ven iz intervala?

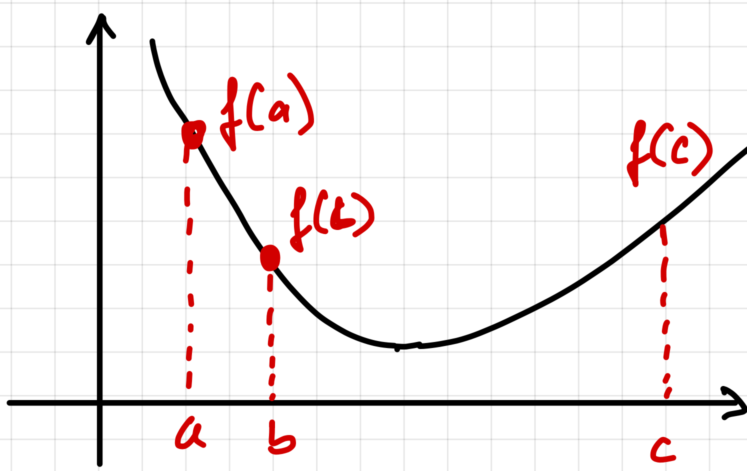


Rekli smo, da algoritem za iskanje ničel deluje za liho število ničel na intervalu $[a,b]$, kjer je $f(a) \cdot f(b) < 0$!

Iskanja lokalnega minimuma se lotimo na podoben način, kot smo se lotili iskanja ničel.

Korak #1: Minimum moramo najprej lokalizirati, (uokviriti). Za to potrebujemo tri točke $a < b < c$, za katere velja, da je

$$f(b) < f(a), f(c)$$



Na takšnem intervalu je vsaj en minimum!

Nekateri algoritmi to danes že sami naredijo za vas! Kljub temu se vedno prepričajte tudi sami, da je na izbranem intervalu res minimum! ⇒ Lahko si narišete funkcijo!

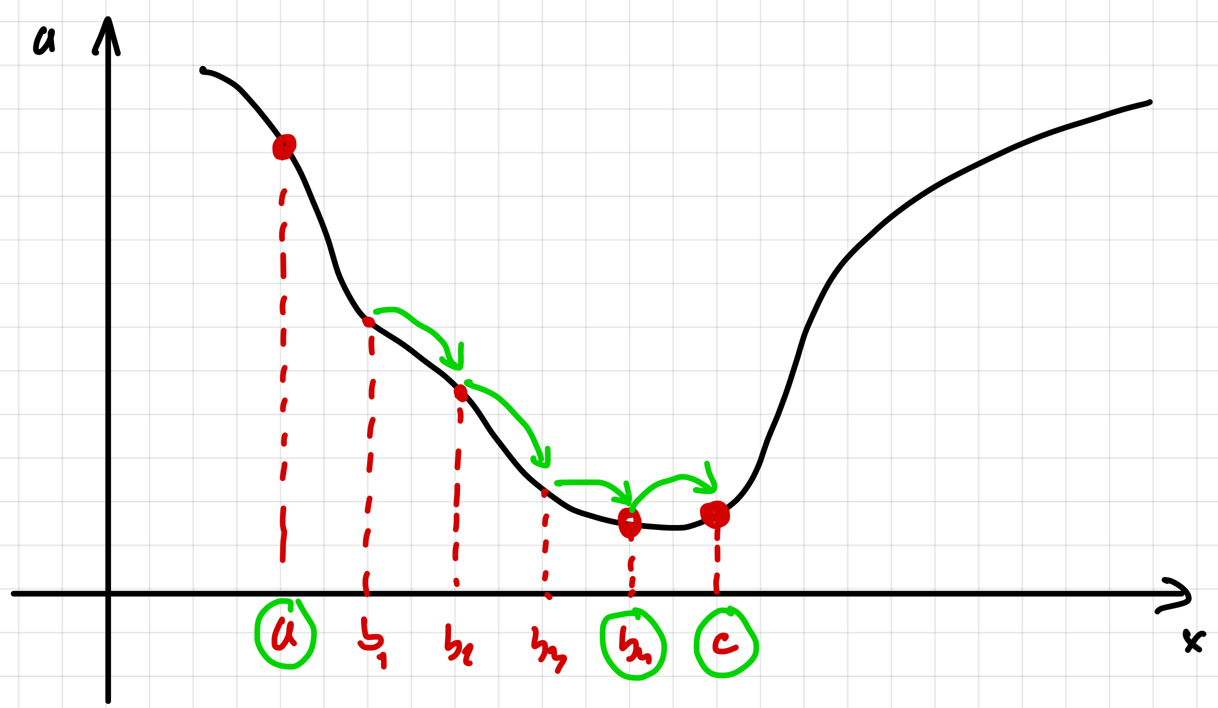
Metoda za uokvirjanje minimuma

1.) Izberi dve točki: a, b, tako da je $f(a) > f(b)$. Če je $f(b) > f(a)$, ju zamenjamo.

2.) Izračunamo tretjo točko $c = b + f \cdot (b-a)$, kjer je f faktor koraka. Korak lahko držimo konstanten ($f=1$), ali pa ga povečujemo v vsakem koraku.

Izračunamo $f(c)$. Če je $f(c) < f(b)$, potem: $b = c$ in ponavljamo korak 2.) vse dokler $f(c) > f(b)$.

3.) Vrni (a, b, c)

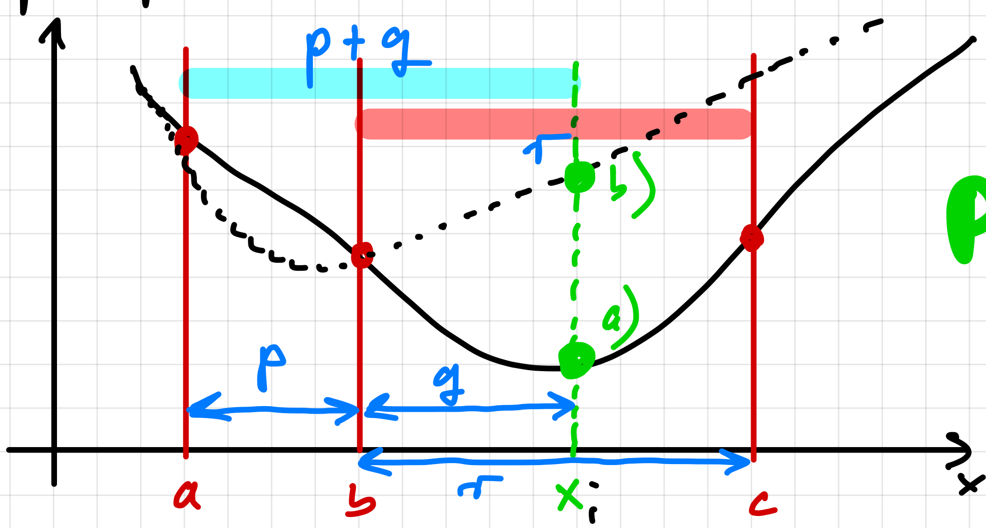


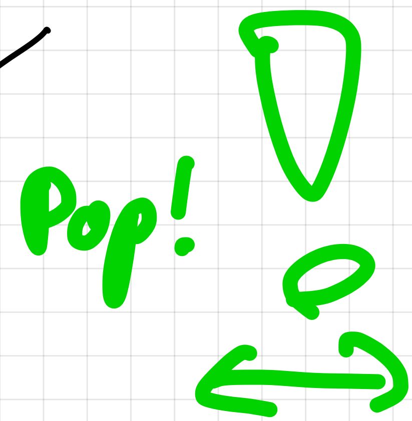
Korak #2:

Ko je minimum uokvirjen, uporabimo algoritem za iskanje minima:

Metoda zlatega reza:

1.) Imamo začetne tri točke: $a < b < c$, kjer je $f(b) < f(a)$ in $f(b) < f(c)$.





2.) Novo točko x_i izberemo na večjem izmed intervalov $[a_i, b_i]$, $[b_i, c_i]$; $i = 0, 1, 2, \ldots$

a.) Če je $f(x_i) < f(b_i)$, potem je naslednja trojka točk $(a_{i+1}, b_{i+1}, c_{i+1})$:

a.)
$$a_{i+1} = b_i$$
$$b_{i+1} = x_i$$
$$c_{i+1} = c_i$$

b.) Če $f(x_i) > f(b_i)$, potem je naslednja trojka točk:

$$
b.) \quad \begin{aligned} a_{i+1} &= a_i \\ b_{i+1} &= b_i \\ c_{i+1} &= x_i \end{aligned}
$$

3.) Ponavljamo 2.) vse dokler

$$
|c_{i+1} - a_{i+1}| < \underline{\underline{\varepsilon}} \gtrsim 10^{-8}!
$$

Izbrana natančnost

Kako izboljšati strategijo določevanja naslednjega približka x_i:

Izberemo:

$$
p = b - a
$$

$$
q = x - b
$$

$$
r = c - b
$$

$\varphi = \dfrac{r}{p}$; Razmerje para, kje je b glede na a ali c!

Točko x_i želimo glede na točke (a_i, b_i, c_i) postaviti tako, da bo konvergenca v naslednjih korakih potekala enako hitro, ne glede na to, ali bomo v naslednjem koraku izbrali trojko a.) ali b.). To zagotovimo tako, sta naslednji intervala enako široka:

$$\overset{b.)}{p+g} = \overset{a.)}{r}$$

Na enak način, kot v i-tem koraku določamo lego x_i, smo v prejšnjem koraku določili lego b_i glede na a_i in c_i. Algoritem skuša, da so razmerja med točkami (a_i, b_i, c_i) enaka razmerjem naslednjih trojk: bodisi (a_i, b_i, x_i) bodisi (b_i, x_i, c_i). Z ohranjanjem razmerij algoritem skuša, da se b_i ne približa preveč a_i oziroma c_i, kar bi lahko znatno upočasnilo konvergenco:

V primeru a.) zato zahtevamo:

$$\underset{\text{i-ti korak}}{\frac{p}{r}} = \underset{\text{razmerje intervalov v i+1. koraku.}}{\frac{g}{r-g}} \quad \left(= \frac{\text{krajši del}}{\text{Dolžina dela}} \right)$$

V primeru b.):

$$\underbrace{\frac{p}{r}}_{i\text{-ti korak}} = \underbrace{\frac{q}{p}}_{i+1\text{-ti korak}} \Rightarrow q = \frac{p^2}{r} \quad (*)$$

Od tod si sedaj lahko izpeljemo:

$$\underbrace{\frac{r}{p}}_{\varphi} = \frac{r-q}{q} = \frac{r}{q} - 1 = \underbrace{\frac{r^2}{q^2}}_{\varphi^2} - 1$$

Dobimo:

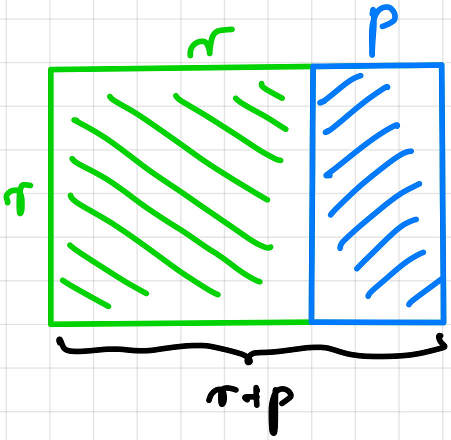
$$\varphi^2 - \varphi - 1 = 0$$

Rešitev te enačbe je:

$$\varphi = \frac{1+\sqrt{5}}{2} = 1.618\ldots..$$

To imenujemo število zlatega reza.

Še malo razmislimo:



$$\varphi^2 = \varphi + 1$$

$$\varphi = 1 + \frac{1}{\varphi}$$

$$\frac{r}{p} = 1 + \frac{p}{r} =$$

$$\boxed{\frac{r}{p} = \frac{p+r}{r}}$$

Zlati rez! $(*)$

Ugotovimo še:

$$\underline{\underline{p+g}} = p + \overset{(*)}{\frac{p^2}{r}} = p\cdot\left(1+\frac{1}{q}\right) = p\,\overset{\text{iz enačbe.}}{\frac{1+q}{q}} =$$

$$= p\cdot\frac{q^2}{q} = p\cdot q = p\cdot\frac{r}{p} = \underline{\underline{r}}$$

Ugotovili smo, da je b_i oddaljena od c_i ravno toliko, kot x_i od a_i. Zato dobimo recept:

1) Če vemo, da je na intervalu $[a_i, c_i]$ minimum, izračunamo:

$$b_i = c_i - r = c_i - \frac{(c_i - a_i)}{q}$$

$$x_i = a_i + r = a_i + \frac{(c_i - a_i)}{q}$$

Komentar:

$$(c-a) = p+r \overset{(*)}{=} \frac{r^2}{p}$$

$$= r\cdot\frac{r}{p} = r\cdot q$$

$$\Downarrow$$

$$r = \frac{c-a}{q}$$

Primerjamo $f(x_i)$ in $f(b_i)$ ter določimo novo trojko (a_i, b_i, c_i).

Ponavljamo, dokler ne dosežemo željene natančnosti! <u>Glej zgled!</u>

Konvergenca: $r = \frac{1}{\varphi} \cdot (c-a) \doteq 0.618.$

Širina intervala v naslednji iteraciji je 0.62 prejšnje, kar je počasneje, kot smo imeli pri bisekciji (0.5). Konvergenca je linearna!

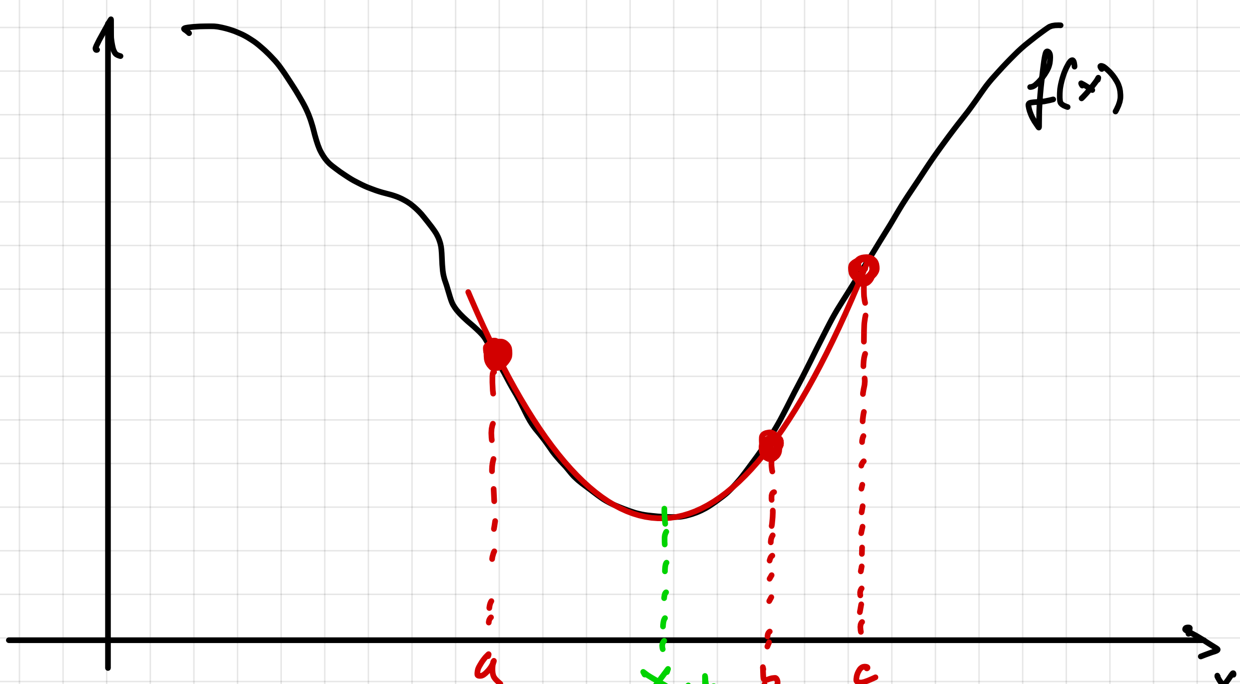
- Metoda zlatega reza je robustna metoda, ki vedno deluje, a nekoliko počasneje! (Analog bisekcije)

- **Ali se da iskanje minimuma pohitriti?** (Da: Metoda z inverzno parabolično interpolacijo, Brentova metoda, ...)

Metoda z inverzno parabolično interpolacijo

Ideja: Taylorjev razvoj pove, da bo funkcija (če je le dovolj pohlevna) v neposredni bližini minimuma imela praktično obliko parabole. To nas napelje na idejo, da če bi tri točke (a, b, c) v tej okolici interpolirali (povezali) s parabolo, bi v eni iteraciji zadeli minimum!

Če pa smo v nekoliko "širši" okolici, pa metoda obljublja hitrejšo konvergenco k minimumu. Metoda ima super linearno konvergenco z ε^n; $n \sim 1.325$.



Poiščimo parabolo $f(x) = Ax^2 + Bx + C$, ki gre skozi točke (a, b, c)

$$f(x) = A(x-b)^2 + B(x-b) + f(b)$$

$$f(c) = A(c-b)^2 + B(c-b) + f(b)$$

$$f(a) = A(a-b)^2 + B(a-b) + f(b)$$

$$f(a) - f(b) = A(a-b)^2 + B(a-b) \quad /:(a-b) \qquad /:(a-b)^2$$

$$f(c) - f(b) = A(c-b)^2 + B(c-b) \quad /:(c-b) \qquad /:(c-b)^2$$

$$\left.\begin{aligned} \frac{f(a)-f(b)}{(a-b)} &= A(a-b) + B \\ \frac{f(c)-f(b)}{(c-b)} &= A(c-b) + B \end{aligned}\right] -$$

$$\frac{f(a)-f(b)}{(a-b)} - \frac{f(c)-f(b)}{(c-b)} = A(a \cancel{-b} - c \cancel{+b}) = A(a-c) \quad /:(-1)$$

$$A = \frac{f(c)-f(b)}{(c-b)(c-a)} + \frac{f(a)-f(b)}{(b-a)\cdot(c-a)}$$

$$\left.\begin{aligned} \frac{f(a)-f(b)}{(a-b)^2} &= A + \frac{B}{a-b} \\ \frac{f(c)-f(b)}{(c-b)^2} &= A + \frac{B}{c-b} \end{aligned}\right] -$$

$$\frac{f(a)-f(b)}{(a-b)^2} - \frac{f(c)-f(b)}{(c-b)^2} = B\left[\frac{1}{a-b} - \frac{1}{c-b}\right]$$

$$= B\,\frac{c \cancel{-b} - a \cancel{+b}}{(a-b)(c-b)} = B\,\frac{(c-a)}{(a-b)(c-b)}$$

$$B = \frac{[f(a)-f(b)](c-b)}{(a-b)(c-a)} - \frac{[f(c)-f(b)](a-b)}{(c-b)(c-a)}$$

Zanima nas, kje ima parabola minimum:

$$f(x) = A(x-b)^2 + B(x-b) + f(c)$$

$$f'(x)\Big|_{x=x_{min}} = 2A(x-b) + B = 0$$

$$x_{min} = b - \frac{B}{2A}$$

Dobimo:

$$x_{min} = b - \frac{1}{2}\,\frac{\dfrac{[f(a)-f(b)](c-b)}{(a-b)(c-a)} - \dfrac{[f(c)-f(b)](a-b)}{(c-b)(c-a)}}{\dfrac{f(a)-f(b)}{(c-b)(c-a)} + \dfrac{f(a)-f(b)}{(b-a)(c-a)}}$$

Znebimo se dvojnih ulomkov z množenjem z $(c-b)(c-a)(b-a)$:

$$x_{min} = b - \frac{1}{2}\,\frac{[f(b)-f(c)](b-a)^2 - [f(b)-f(a)](b-c)^2}{[f(b)-f(c)](b-a) - [f(b)-f(a)](b-c)}$$

Postopek:

1.) Določimo $a < b < c$, da je $f(b) < f(a), f(c)$

2.) Izračunamo x_{min} in $f(x_{min})$

- Če je $f(x_{min}) < f(b)$ in $x_{min} < b$!

  $c = b$ in $b = x_{min}$

- Če je $f(x_{min}) < f(b)$ in $x_{min} > b$!

  $a = b$ in $b = x_{min}$

- Če je $f(x_{min}) > f(b)$ in $x_{min} < b$!

  $a = x_{min}$

- Če je $f(x_{min}) > f(b)$ in $x_{min} > b$:

  $c = x_{min}$.

3.) Ponavljamo, dokler $|a - c| < \underline{\underline{\varepsilon}}$. $\sim 10^{-8}$

Izberem

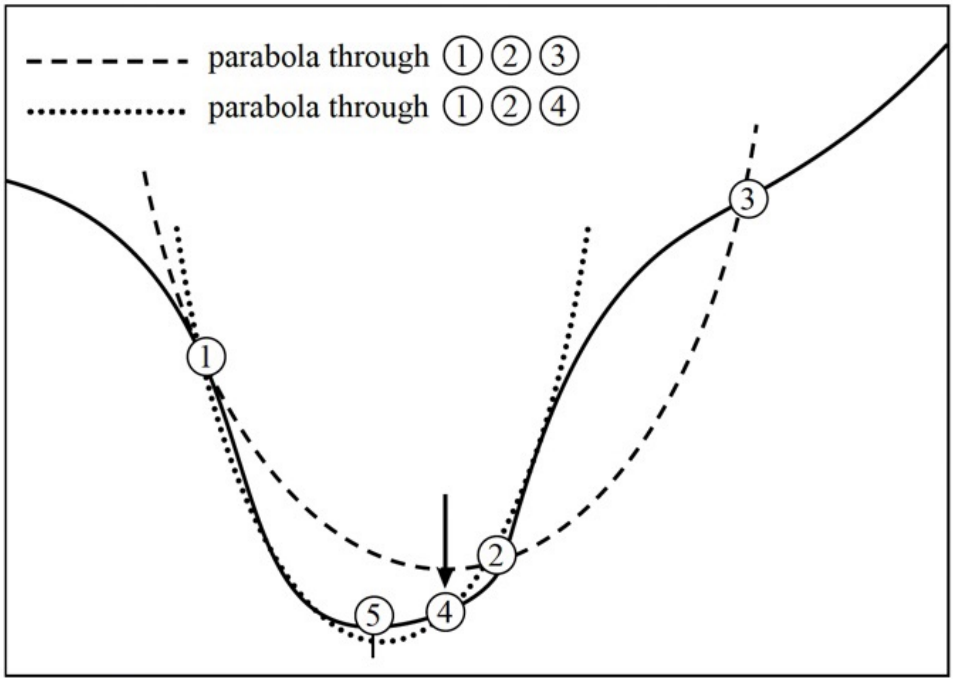


Figure 10.2.1. Convergence to a minimum by inverse parabolic interpolation. A parabola (dashed line) is drawn through the three original points 1,2,3 on the given function (solid line). The function is evaluated at the parabola's minimum, 4, which replaces point 3. A new parabola (dotted line) is drawn through points 1,4,2. The minimum of this parabola is at 5, which is close to the minimum of the function.

Težave:

- Metoda ne loči med minimumom in maksimumom.
- Metoda kot taka ni nujno da konvergira, lahko jo zanese izven začetnega intervala!
- Težave se pojavijo tedaj, ko je funkcija zelo položna. Tedaj je $A \sim 0$ da računanje odvoda ne deluje dobro.

V tej obliki se metoda ne uporablja!

Brentova metoda: (Gleich NRC, str. 402)

Metoda na robusten način združuje hitro parabolično metodo z robustnejšo metodo zlatega reza v primerih, ko metoda ugotovi, da parabolična metoda ne konvergira. To doseže s previdnim knjigovodstvom zadnjih nekaj vrednosti in z njimi preverja, da so naslednje iteracije parabolične metode smiselne. (npr. da naslednji x_n leži znotraj $[a,b]$).

Dosegljiva natančnost pri iskanju minimuma:

Iz izkušnje pri iskanju ničel funkcije, pričakujemo, da bomo z algoritmom minimume ujeli na intervale:

$$|c-a| < \varepsilon$$

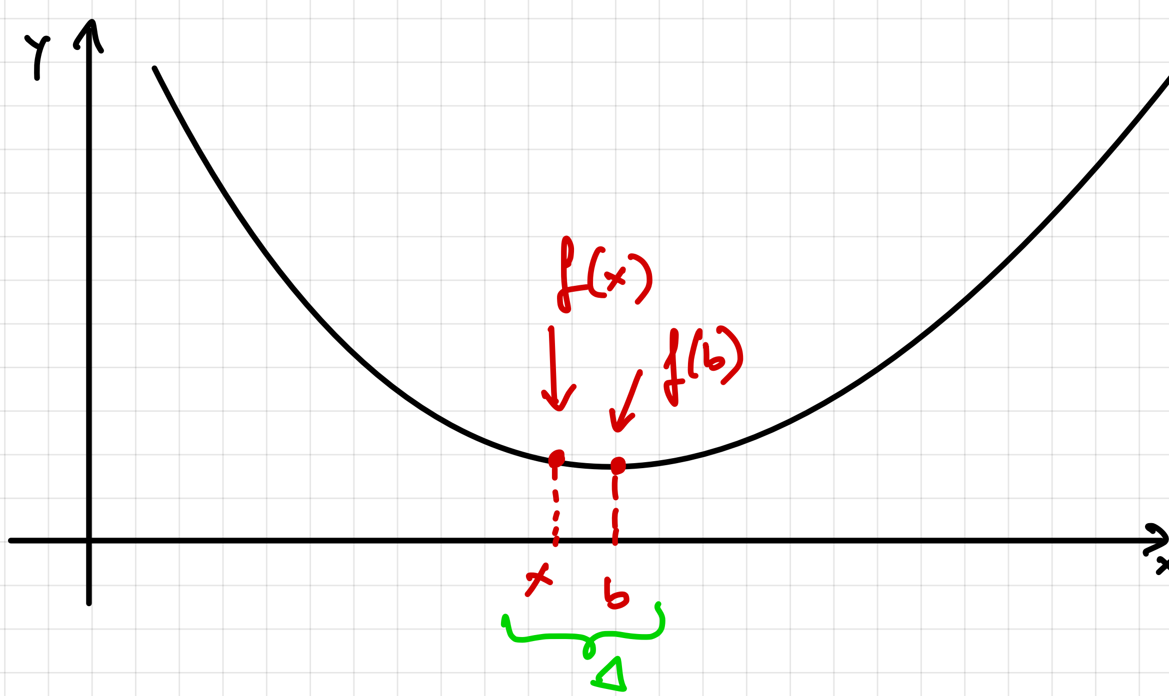
ali

Najboljši približek.

$$(1-\varepsilon)\cdot b \le b \le (1+\varepsilon)\, b$$

kjer je $\varepsilon \sim 10^{-15}$ naša računska natančnost v aritmetiki s plavajočo vejico.

Pozor: Natančnosti $\varepsilon \sim 10^{-15}$ ne bomo nikoli dosegli, saj nas bo prej omejila natančnost, s katero znamo računati vrednosti funkcije v okolici minimuma.



Vsaj koliko morata biti točki x in b narazen ($\Delta = x - b$), da bomo še uspeli ločiti $f(x)$ in $f(b)$?

Vrednost $f(x)$ v okolici $f(b)$ razvijemo po Taylorjevem razvoju:

$$f(x) \simeq f(b) + \underbrace{f'(b)}_{\substack{= 0 \\ \text{(Minimum)}}}(x-b) + \frac{1}{2} \overset{\neq 0}{f''(b)} \cdot (x-b)^2 + \dots$$

<u>Relativno odstopanje:</u>

$$\Bigg| \underbrace{\frac{f(x) - f(b)}{f(b)}}_{\underbrace{\frac{f(x)}{f(b)} - 1}_{\equiv \text{Decimalna mesta}}} \Bigg| \simeq \frac{1}{2} \frac{f''(b)}{|f(b)|} (x-b)^2 \geq \varepsilon$$

$$|x - b| \geq \sqrt{\varepsilon} \cdot \sqrt{\frac{2\,|f(b)|\, b^2}{f''(b)\; b^2}}$$

Pomnožimo da bodo lepše enote

$$\boxed{|x-b| \geq \sqrt{\varepsilon} \cdot b \cdot \sqrt{\frac{2\,|f(b)|}{b^2 f''(b)}}}$$

~ 1

Dobimo:

$$|x-b| \geq \sqrt{\varepsilon}\cdot b$$

$$\left|\frac{x-b}{b}\right| \geq \sqrt{\varepsilon} = \varepsilon_{eff} = 10^{-8}$$

Največja natančnost iskanja minimov je zaradi $f'(b)=0$ približno 10^{-8}. Če bomo zahtevali 10^{-15} algoritem ne bo nikoli konvergiral!

Zgled: Izračunaj minimum funkcije $f(x) = x + \frac{1}{x}$, če $x \in (0, \infty)$.

Kje ima funkcija ekstrem: $f'(x) = 0$?

$$f'(x) = 1 - \frac{1}{x^2} = 0$$

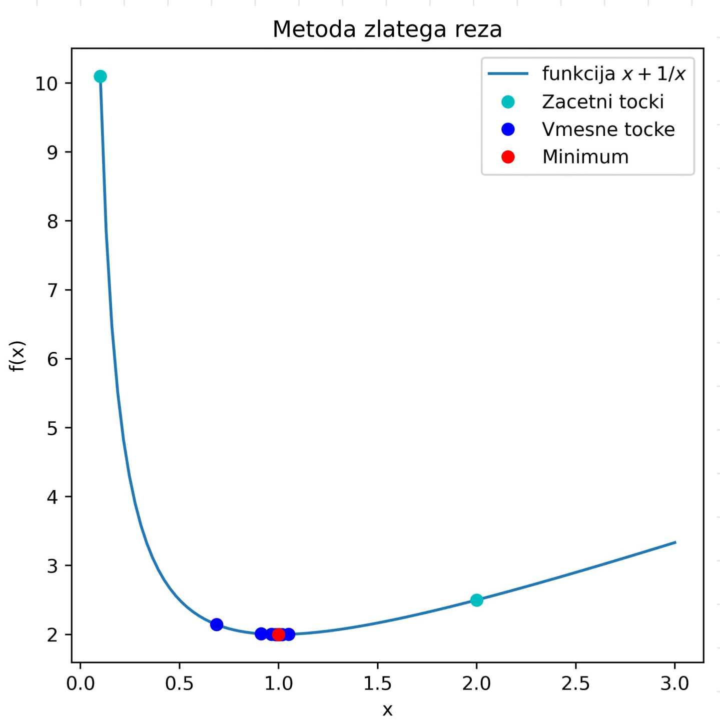
$$x^2 = 1$$

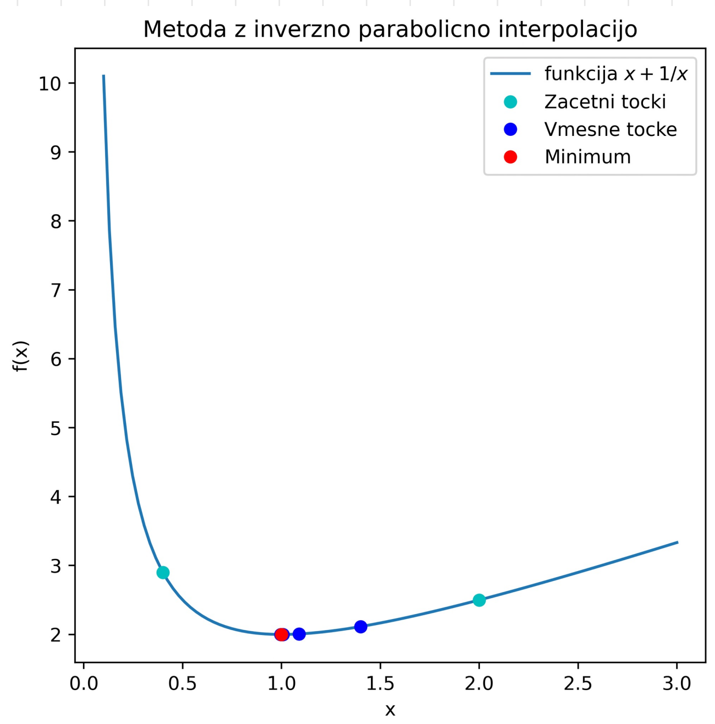
$$x = \pm\sqrt{1} = 1$$

To je minimum funkcije!

Kaj pa drugi odvod?

$$f''(x) = \frac{2}{x^3}. \qquad f''(x=1) = 2 > 0$$





Minimizacija v več dimenzijah

Za izvedbo minimizacije obstaja vrsta različnih algoritmov: Ameba, Powellova metoda konjugiranih smeri, metoda konjugiranih gradientov, metode s spremenljivo metriko, metoda simuliranega ohlajanja.

Ameba (ali metoda Nelder–Mead ali "Downhill simplex" metoda)

- To je najbolj splošno uporabljena metoda
- Velika prednost metode je, da zahteva le vrednosti funkcije. Za delovanje ne potrebuje vrednosti odvodov.

Osnovni objekt metode je simpleks, geometrijski objekt, ki ima v N dimenzionalnem prostoru N+1 oglišč, ki jih povezujejo stranice, mnogokotniki itd.

V dvodimenzionalnem prostoru (N=2) je simpleks trikotnik.

Opomba: Delovanje metode si bomo pogledali na primeru $n=2$, saj si tu reči najlažje predstavljamo, hkrati pa ne izgubljamo splošnosti.

Omejitev: Pri enodimenzionalni minimizaciji smo minimum uokvirili/lokalizirali in si s tem zagotovili, da bomo minimum našli! Tega pri večdimenzionalni minimizaciji ni! Vse kar lahko naredimo je, da algoritmu damo začetni približek, nato pa pustimo, da gre algoritem svojo pot po prostoru dokler ne zadane lokalnega minimuma.

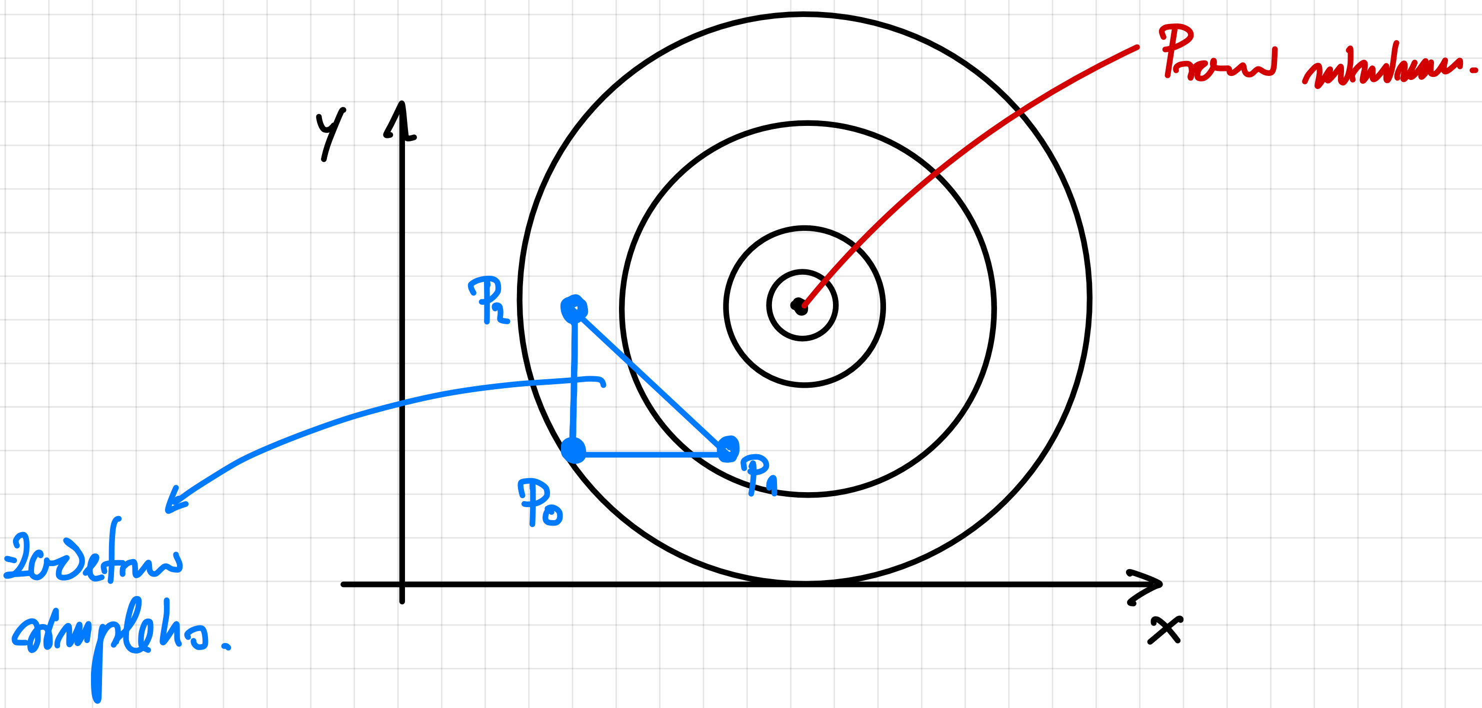
Začetni približek: Algoritem zahteva $n+1$ začetnih točk, ki definirajo začetni simpleks. Tega lahko konstruiramo na naslednji način:

- Izberemo eno točko $\vec{P}_0 = (x_0, y_0)$

- Ostala oglišča simpleksa določimo tako, da se premaknemo vzdolž enotskih vektorjev prostora:

$$\vec{P}_i = \vec{P}_0 + \lambda_i \cdot \hat{e}_i \quad ; \quad i = 1, \dots, n.$$

Pri tem so λ_i tipične dolžinske skale na katerih se splača iskati minimum.
Vrednosti λ_i so odvisne od primera, zato moramo imeti nek uvid v problem!
Ne uporabljaj algoritma na slepo!

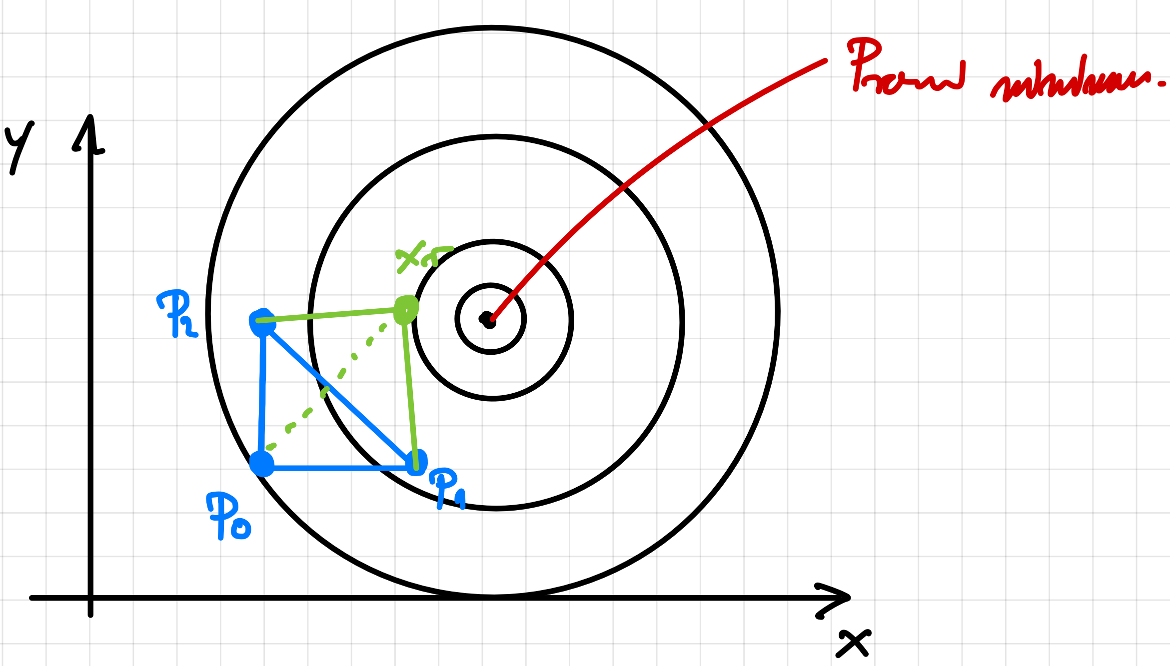


Iskanje minimuma je niz zaporednih transformacij s katerimi začetni simpleks najde pot do lokalnega minimuma in se "zasede vanj."

1. korak: Oglišča simpleksa uredimo po velikosti:
$$f(x_1) \leq f(x_2) \leq f(x_3)$$

2. korak: Transformiramo simpleks. Možne so naslednje transformacije:

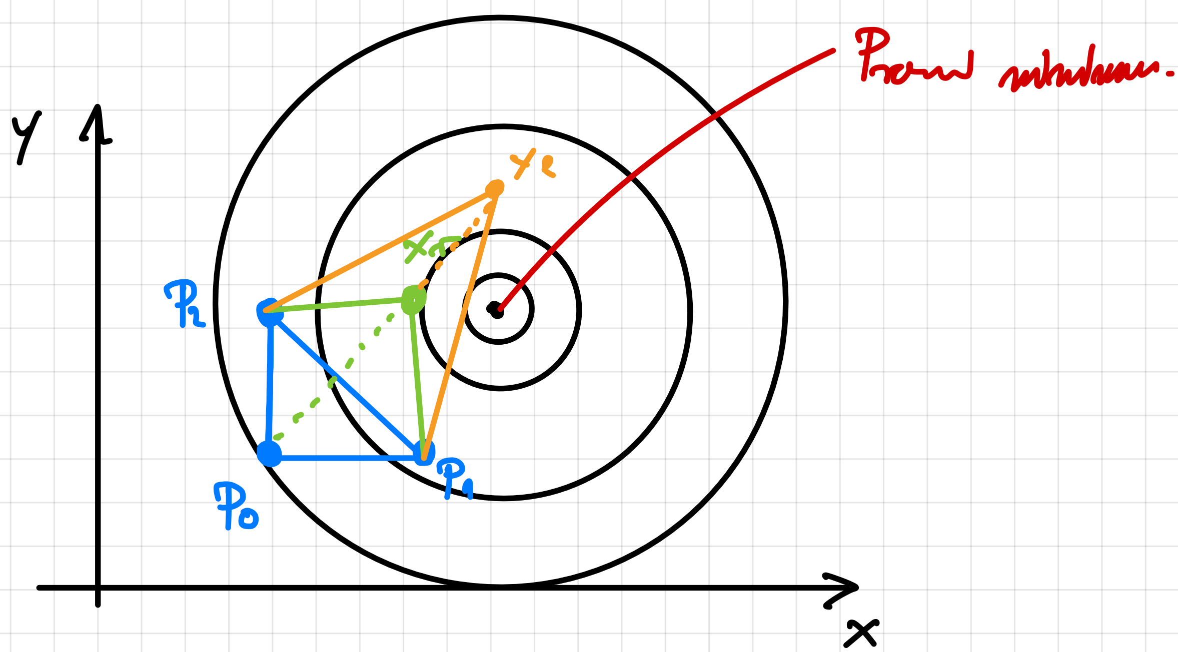
Zrcaljenje: To transformacijo naredimo najprej. Zrcalimo najslabšo točko: $f(x_3)$ preko nasprotne ležne stranice. Dobimo x_r in $f(x_r)$



- Če je $f(x_1) < f(x_r) < f(x_2) < f(x_3)$ potem sprejmemo točko $(x_3 = x_r)$ in se vrnemo na 1.)
- Če je $f(x_r) < f(x_1) < f(x_2) < f(x_3)$, potem izvedemo še razteg.
- Če je $f(x_r) > f(x_2)$, potem izvedemo še kontrakcijo.

Razteg: Simpleks raztegnemo v smer zrcaljenja za $\gamma > 1$ in dobimo točko x_e, $f(x_e)$

- Če je $f(x_e) < f(x_r)$ potem $x_3 = x_e$
- Če je $f(x_e) > f(x_r)$ potem $x_3 = x_r$

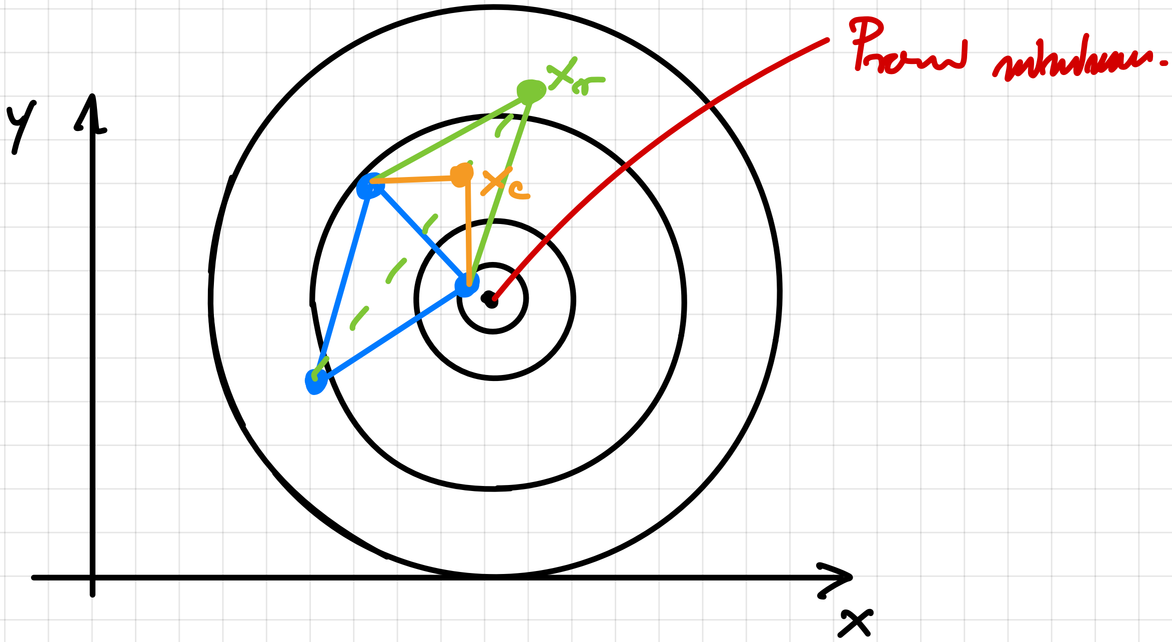


Ožanje: Tu je $f(x_r) \geq f(x_2)$

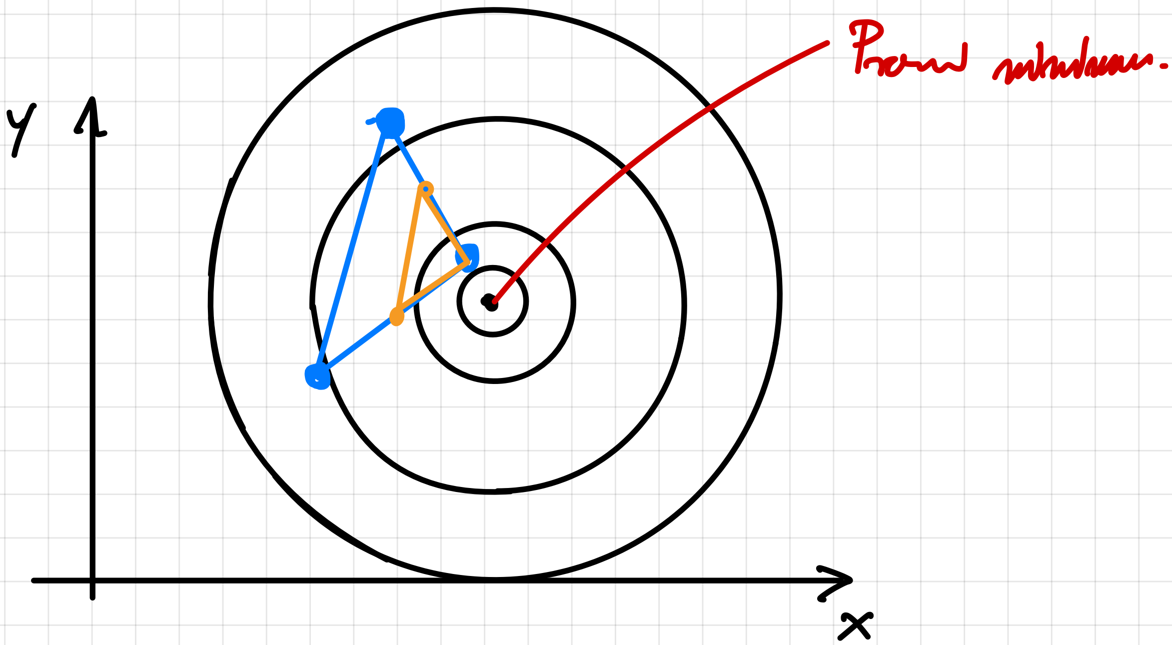
Druga najslabša točka

- Če je $f(x_r) < f(x_3)$ potem prezrcaljeni trikotnik zožamo (razteg s faktorjem $0 < \rho < \frac{1}{2}$) v smer zrcaljenja. Dobimo točko x_c in $f(x_c)$. Če je $f(x_c) < f(x_r)$ potem sprejmemo točko $x_3 = x_c$ in se vrnemo na 1.) Če ožanje ne pomaga ($f(x_c) > f(x_r)$), potem izvedi še krčenje

- Če je $f(x_r) > f(x_3)$ potem prezrcaljeni trikotnik zožaj za faktor $0 < \rho < \frac{1}{2}$. Dobiš točko x_c, $f(x_c)$. Če je $f(x_c) < f(x_3)$, potem sprejmi točko $x_3 = x_c$ in pojdi na 1.). Če ožanje ne pomaga potem pojdi na krčenje.



Krčenje: Simpleks skrčimo tako, da vse točke razen najboljše premaknemo bližje tej točki za faktor $\sigma \sim \frac{1}{2}$ vzdolž daljic, ki povezujejo oglišča z najboljšim.



3. korak:

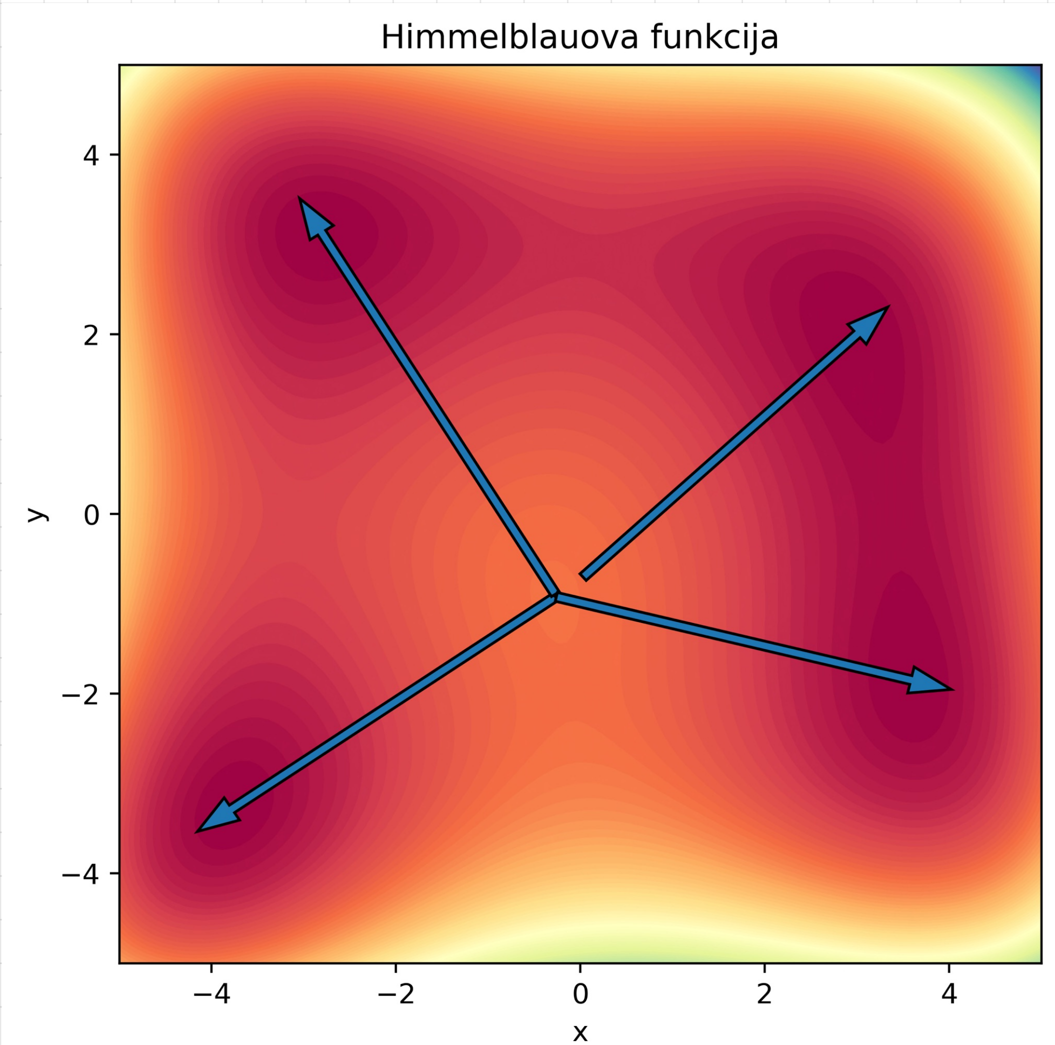
Minimizacijo zaključimo, ko je razlika med najboljšo in najslabšo točko manjša od izbrane natančnosti:

$$|f(x_1) - f(x_3)| < \varepsilon \sim \underline{10^{-8}}$$

Za minimum izberemo najboljšo točko x_1.

Zgled: Poišči minimum(e) Himmelblauove funkcije:

$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$



Funkcija ima 4 različne minimume. Metoda konvergira v enega izmed njih, odvisno od izbire začetnih pogojev. Že zelo majhne spremembe lahko vodijo do zelo različnih rezultatov.

Prilagajanje funkcij podatkom

- Prilagajanje funkcij podatkom je pomemben primer uporabe minimizacijskih algoritmov – predvsem tistih za minimizacijo v več dimenzijah.

Imamo nabor merskih podatkov:

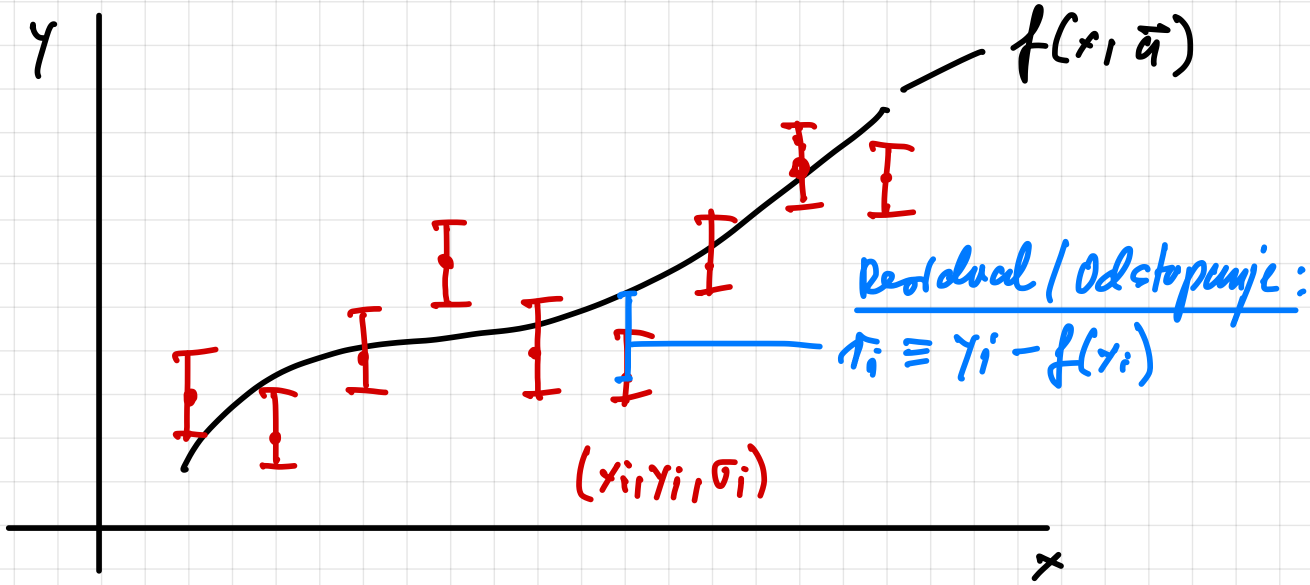
$$\{(\underbrace{x_i}_{\text{neodvisna spremenljivka}}, \underbrace{y_i}_{\text{vrednost funkcije}}, \underbrace{\sigma_i}_{\text{nedoločenost meritve.}})\}_{i=1}^{N}$$

N – število merskih točk.

Radi bi ugotovili, kako je y odvisen od x $\Rightarrow$ Izvesti želimo regresijo odvisne spremenljivke y po neodvisni spremenljivki $\equiv$ podatkom prilagodimo regresijsko krivuljo

$$f(x, \vec{a}), \qquad \underbrace{\vec{a} = (a_1, \dots, a_m)}_{\text{Parametri krivulje.}}$$

Funkcija $f(x, \vec{a})$ je model za podatke. Prilagajanje (oz. "Fitanje") pa pomeni iskanje takšnih parametrov, da se model kar najbolje prilega podatkom.



Obstaja več kriterijev za to, kaj je "dobro ujemanje". Najpogosteje uporabljamo metodo najmanjših kvadratov:

V tej metodi odstopanje modelske krivulje od podatkov ocenjujemo s cenilno funkcijo χ^2:

$$\chi^2(\vec a) = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i)}{\sigma_i}\right)^2 = \underbrace{\sum_{i=1}^{N} \left(\frac{r_i}{\sigma_i}\right)^2}$$

Kako dobro se krivulja prilega podatkom merimo z vsoto kvadratov relativnih odstopanj točk.

Cenilka $\chi^2(\vec{a})$ je funkcija parametrov $\vec{a}$ modela $f(x, \vec{a})$. Model se bo v okviru izbrane metode najbolje prilegal podatkom pri tistem naboru parametrov $\vec{a}'$, kjer je

$$\chi^2(\vec{a}') = \text{min.}$$

Iskanje parametrov $\vec{a}' = (a_1', a_2', a_3', \ldots a_n')$ kjer je χ^2 minimalna izvedemo z minimizacijskim algoritmom, npr. Amebo!

$\Downarrow$

Če uspemo najti globalni minimum za χ^2, potem vemo, da se izbrani model pri teh parametrih v smislu najmanjših kvadratov najbolje prilega izbranim podatkom.

(Pri tem seveda še vedno obstaja možnost, da obstaja neka druga funkcija $g(x, \vec{b})$ z drugačnimi parametri, ki bi dala še manjšo vrednost χ^2!)

Past: Ko je minimizacija končana, nam nihče ne garantira, da nismo končali v lokalnem minimumu!

Linearna regresija: (komentar)

O linearni regresiji govorimo, kadar je modelska funkcija linearno odvisna od parametrov a_i.

Npr.:

$$f(x, \vec{a}) = \underline{a_1} + \underline{a_2} x + \underline{a_3} x + \cdots + \underline{a_m} x^{m-1}$$

$$f(x, \vec{a}) = \underline{a_1} x + \underline{a_2} e^{x} + \underline{a_3} \sin x + \underline{a_4} \log x$$

$$\vdots$$

$$f(x, \vec{a}) = \sum_{j=1}^{m} \underbrace{a_j}_{\text{parameter}} \cdot \underbrace{\theta(x)}_{\text{Izbrana funkcija } x.}$$

Tudi tu poskušamo minimizirati χ^2:

$$\frac{\partial \chi^2}{\partial a_j} = 0 \quad ; \quad j = 1 \dots m$$

Od tod dobimo linearen sistem enačb za parametre a_j, ki ga znamo rešiti (Npr. SVD). Dobimo rešitev, ki pa je v smislu najmanjših kvadratov gotovo najboljša!

Glej: Širca, VF, str. 197.

Nasvet: Vaš problem vedno poskušajte linearizirati. **Npr:** Prilagajamo

$$f(x;a,b) = a e^{-bx}$$

Uvedemo: $g(x;a,b) = \ln(f(x))$

$$g(x;a,b) = \underbrace{\ln a}_{a'} - bx = a' - bx$$

Problem smo linearizirali!

Prilagajanje premice:

Prilagajanje premice je primer linearne regresije za $m = 2$.

$$f(x;\ \underbrace{a,\ b}_{\text{Dva parametra.}}) = ax + b$$

Minimiziramo:

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

Računamo:

$$\frac{\partial \chi^2}{\partial a} = -2 \sum_{i=1}^{n} \frac{(y_i - a x_i - b)\, x_i}{\sigma_i^2} = 0$$

$$\frac{\partial \chi^2}{\partial b} = -2 \sum_{i=1}^{n} \frac{(y_i - a x_i - b)}{\sigma_i^2} = 0$$

Če vpeljemo:

$$\left.\begin{aligned}
A &= \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2}, \quad & B &= \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \\
C &= \sum_{i=1}^{n} \frac{y_i}{\sigma_i^2}, \quad & D &= \sum_{i=1}^{n} \frac{x_i^2}{\sigma_i^2} \\
E &= \sum_{i=1}^{n} \frac{x_i y_i}{\sigma_i^2}, \quad & F &= \sum_{i=1}^{n} \frac{y_i^2}{\sigma_i^2}
\end{aligned}\right\}$$

To znamo izračunati iz podatkov!

dobimo sistem enačb:

$$aA + bB = C$$
$$aD + bA = E$$

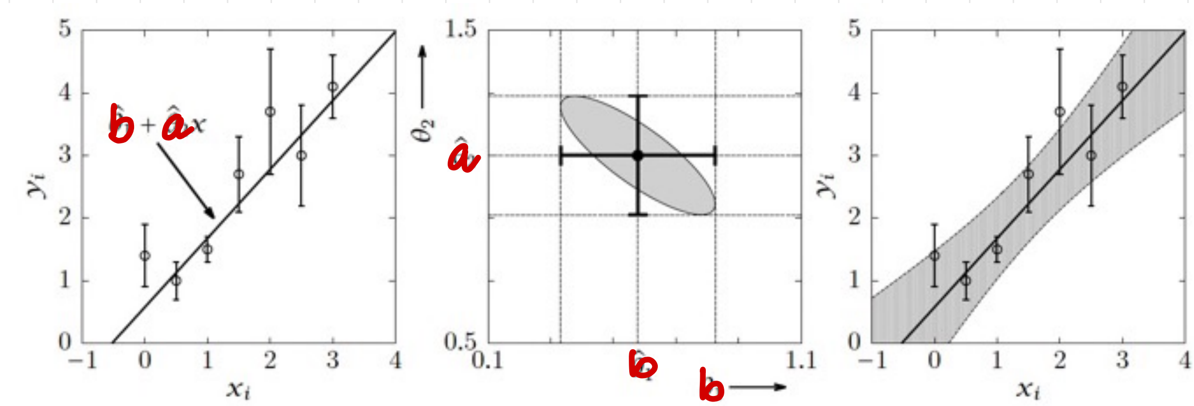
in rešitve:

$$a = \frac{EB - CA}{DB - A^2}, \quad b = \frac{DC - EA}{DB - A^2}$$

Izračunati se da tudi napake (variance) parametrov in korelacijski koeficient (Glej, Širca, VF, 206)

$$\sigma_a^2 = \frac{B}{BD - A^2}, \quad \sigma_b^2 = \frac{D}{BD - A^2}$$

$$\mathrm{cov}(a,b) = \frac{-A}{BD - A^2}, \quad \rho(a,b) = \frac{\mathrm{cov}(a,b)}{\sigma_a \cdot \sigma_b}$$



Slika 9.3 — Prilagajanje linearne funkcije izmerkom $y_i = 1.4, 1.0, 1.5, 2.7, 3.7, 3.0, 4.1$ z napakami $\sigma_i = 0.5, 0.3, 0.2, 0.6, 1.0, 0.8, 0.5$ v točkah $x_i = 0, 0.5, 1, 1.5, 2, 2.5, 3$ po metodi najmanjših kvadratov. [LEVO] Premica, ki minimizira mero odstopanja X^2 ($\hat{\theta}_1 = 0.578$, $\hat{\theta}_2 = 1.100$). [SREDINA] Kovariančna elipsa s središčem v točki $(\hat{\theta}_1, \hat{\theta}_2)$ in napakama $\pm\sigma(\hat{\theta}_1) = \pm 0.247$ in $\pm\sigma(\hat{\theta}_2) = \pm 0.190$. Parametrom iz notranjosti elipse ustrezajo premice, ki vse potekajo po senčenem območju na sliki [DESNO]. Premica, ki ustreza pravima vrednostma θ_1 in θ_2, z verjetnostjo $1 - e^{-1/2}$ leži znotraj te pahljače.

Zgled:

(Leo, str 99)

x	0	1	2	3	4	5
y	0.92	4.15	9.78	14.46	17.26	21.90
σ	0.5	1.0	0.75	1.25	1.0	1.5

Podatkom iz tabele prilagodi premico $f(x) = ax + b$.

Izračunane vrednosti:

$$a = 4.227 \pm 0.044$$

$$b = 0.878 \pm 0.203$$

$$\chi^2 = 2.078$$

$$\mathrm{cov}(a,b) = -0.0629$$

$$\delta = 7.04$$

Rezultat minimizacije z Amebo v Pythonu: