Optimum számítási módszerek

Papp Zoltán

Mechatronika MSc

Optimalizálási probléma

Optimalizálás

egy függvény (a célfüggvény) minimalizálása vagy maximalizálása

Optimalizálási probléma

Optimalizálás

egy függvény (a célfüggvény) minimalizálása vagy maximalizálása

Feltétel nélküli optimalizálási probléma

nincsenek kikötések a célfüggvény változóira

$$\min_{x \in R^n} f(x)$$

Optimalizálási probléma

Optimalizálás

egy függvény (a célfüggvény) minimalizálása vagy maximalizálása

Feltétel nélküli optimalizálási probléma

nincsenek kikötések a célfüggvény változóira

$$\min_{x \in R^n} f(x)$$

Feltételes optimalizálási probléma

megengedett megoldások halmazával, $\Omega \subset R^n$, megadott feltételek a célfüggvény változóira

$$\min_{x \in \Omega} f(x)$$

Optimalizálási probléma

Optimalizálás

egy függvény (a célfüggvény) minimalizálása vagy maximalizálása

Feltétel nélküli optimalizálási probléma

nincsenek kikötések a célfüggvény változóira

$$\min_{x \in R^n} f(x)$$

Feltételes optimalizálási probléma

megengedett megoldások halmazával, $\Omega \subset R^n$, megadott feltételek a célfüggvény változóira

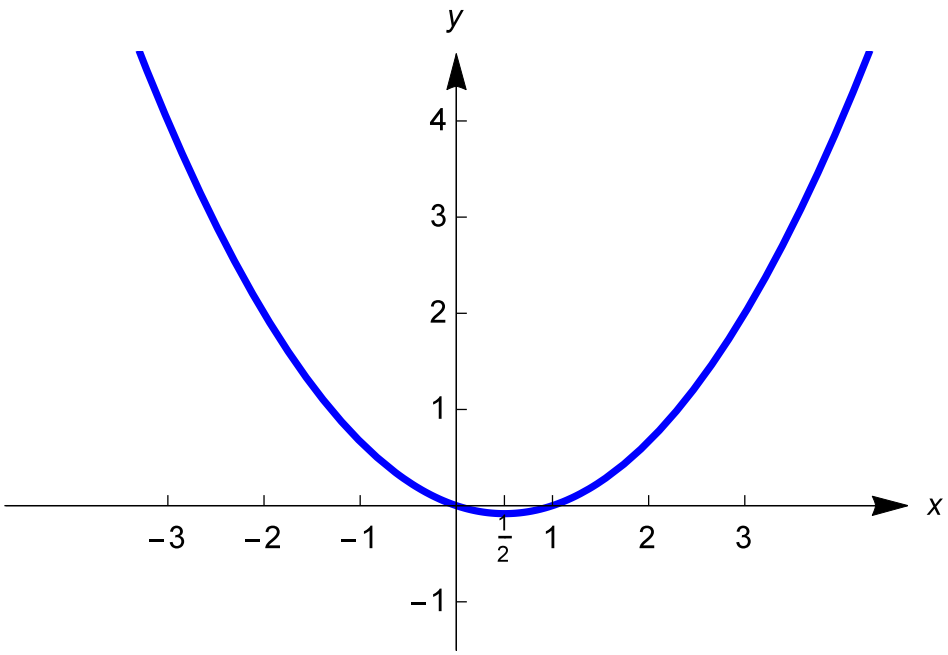
$$\min_{x \in \Omega} f(x)$$

$$\Omega = \{x : g_i(x) = 0, i = 1, 2, \ldots, m, \; g_i(x) \geq 0, i = m+1, m+2, \ldots, s\}$$

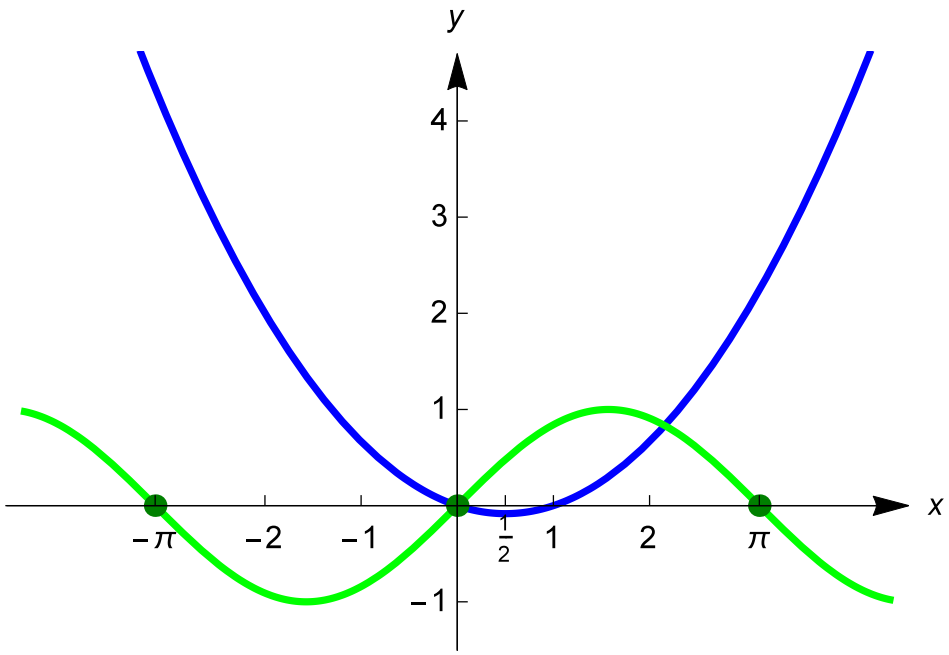
Feltétel nélküli és feltételes minimalizálási feladatok

Példa $\mathbb{R}$-ben

$$\min_{x \in R} \frac{x^2 - x}{3}$$



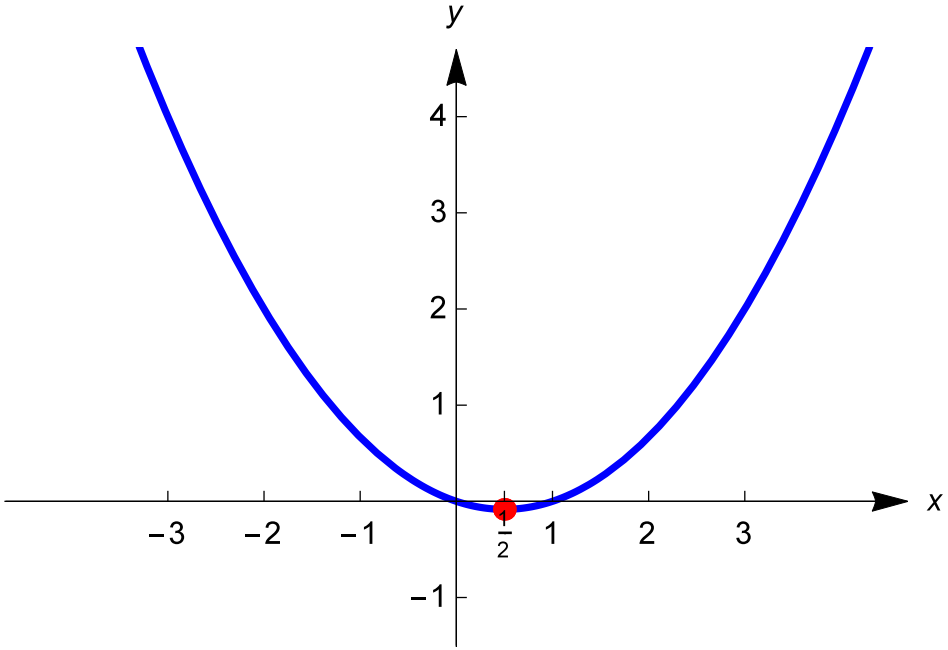
$$\min_{x \in \Omega} \frac{x^2 - x}{3}, \quad \Omega = \{x : \sin x = 0\}$$



Feltétel nélküli és feltételes minimalizálási feladatok

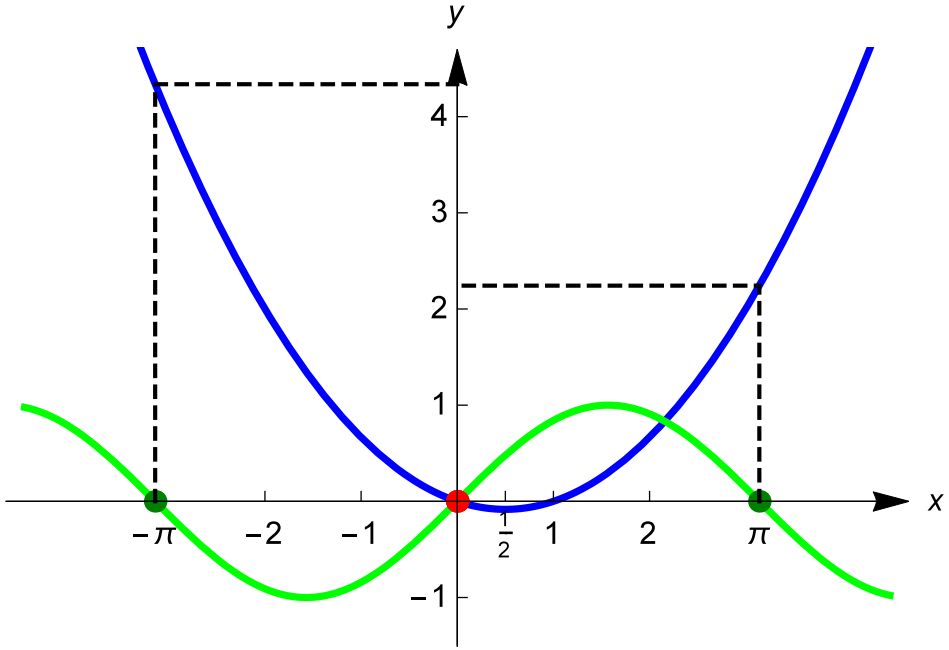
Példa $\mathbb{R}$-ben

$$\min_{x \in R} \frac{x^2 - x}{3}$$



megoldás: $f_{min} = -\frac{1}{12}$, $x = -\frac{1}{2}$

$$\min_{x \in \Omega} \frac{x^2 - x}{3}, \quad \Omega = \{x : \sin x = 0\}$$

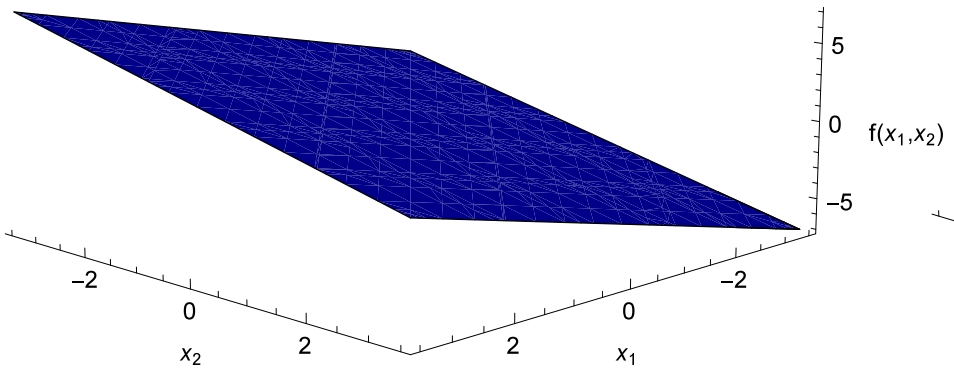


megoldás: $f_{min} = 0$, $x = 0$

Feltétel nélküli és feltételes minimalizálási feladatok

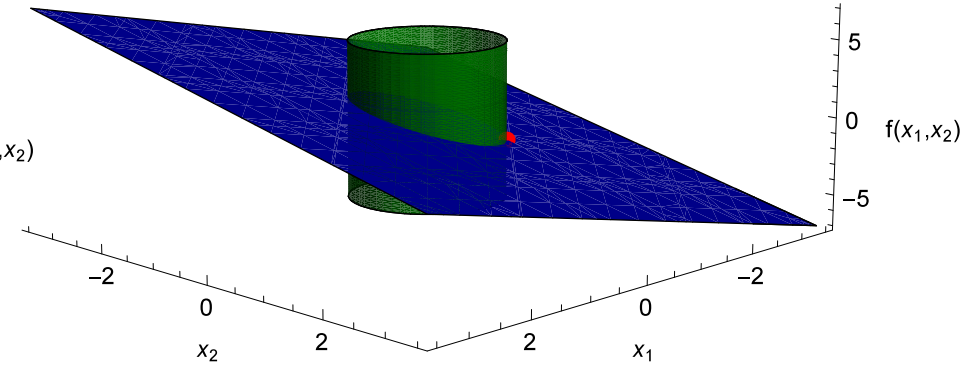
Példa $\mathbb{R}^n$-ben

$$\min_{x \in R^2} x_1 - x_2$$



megoldás: nincs

$$\min_{x \in \Omega} x_1 - x_2, \quad \Omega = \{x : x_1^2 + x_2^2 - 1 = 0\}$$



megoldás:

$f_{min} = -\sqrt{2}$, $x = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$

Optimalizálási probléma

Determinisztikus:

teljesen ismert modell

$$\min_{x \in \Omega} f(x)$$

$$\Omega = \{x : \quad g_i(x) = 0, \ i = 1, 2, \ldots, m, \quad g_i(x) \geq 0, \ i = m+1, m+2, \ldots, s\}$$

Optimalizálási probléma

Determinisztikus:

teljesen ismert modell

$$\min_{x\in\Omega} f(x)$$
$$\Omega = \{x : \quad g_i(x) = 0,\ i = 1, 2, \ldots, m,$$
$$g_i(x) \geq 0,\ i = m+1, m+2, \ldots, s\}$$

Sztochasztikus: a zajok, zavarok, minden létező fizikai rendszer elemi jellemzői

ismeretlen hatások (véletlen függvényértékek vagy/és véletlen input értékek) a modell meghatározásánál $\Rightarrow \xi$ véletlen változó

$$\min_{x\in\Omega} f(x,\xi)$$
$$\Omega = \{x : \quad g_i(x,\xi) = 0,\ i = 1, 2, \ldots, m,$$
$$g_i(x,\xi) \geq 0,\ i = m+1, m+2, \ldots, s\}$$

Mi a megoldása a feltétel nélküli optimalizálási feladatnak?

Definíció

Az $x^* \in \mathbb{R}^n$ *pont az* $f : \mathbb{R}^n \to \mathbb{R}$ *függvény globális minimuma ha* $f(x^*) \leq f(x)$ *minden* $x \in \mathbb{R}^n$ *pontra.*

Mi a megoldása a feltétel nélküli optimalizálási feladatnak?

Definíció

Az $x^* \in \mathbb{R}^n$ *pont az* $f : \mathbb{R}^n \to \mathbb{R}$ *függvény globális minimuma ha* $f(x^*) \leq f(x)$ *minden* $x \in \mathbb{R}^n$ *pontra.*

Definíció

Az $x^* \in \mathbb{R}^n$ *pont az* $f : \mathbb{R}^n \to \mathbb{R}$ *függvény lokális minimuma ha létezik az* x^* *pont olyan* $\mathcal{N}$ *környezete, melyben* $f(x^*) \leq f(x)$ *minden* $x \in \mathcal{N}$ *pontra.*

Mi a megoldása a feltétel nélküli optimalizálási feladatnak?

Definíció

Az $x^* \in \mathbb{R}^n$ *pont az* $f : \mathbb{R}^n \to \mathbb{R}$ *függvény globális minimuma ha* $f(x^*) \leq f(x)$ *minden* $x \in \mathbb{R}^n$ *pontra.*

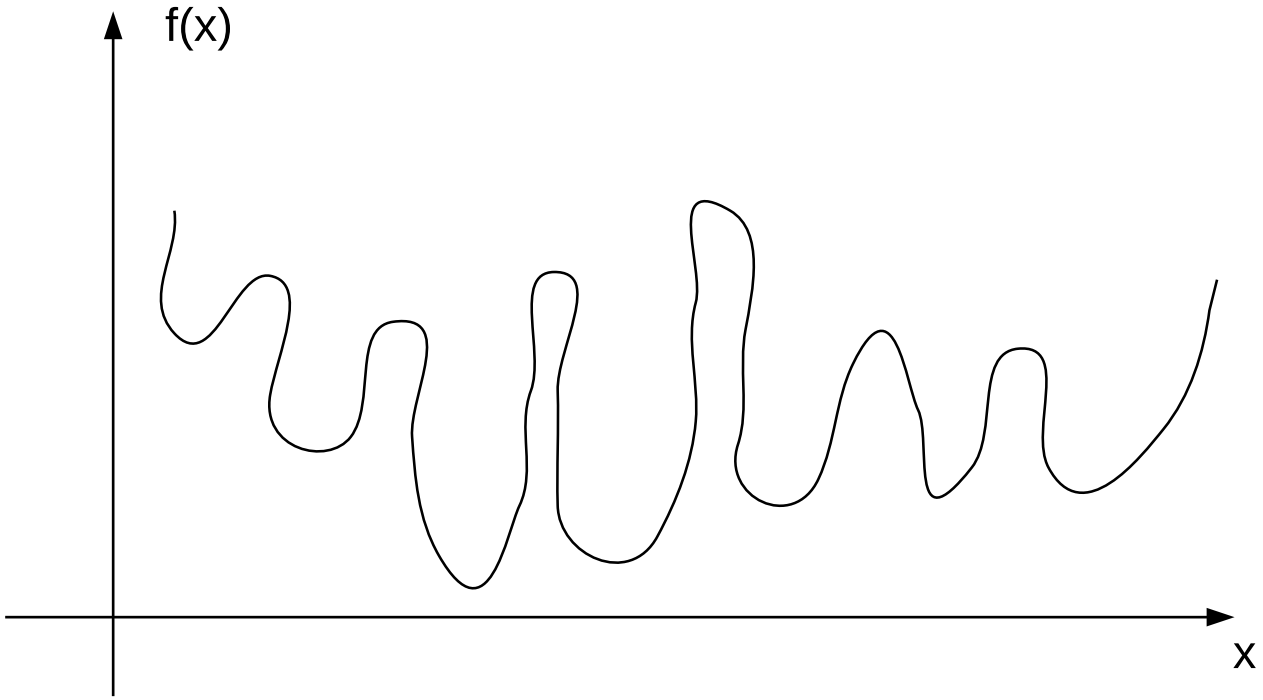
Definíció

Az $x^* \in \mathbb{R}^n$ *pont az* $f : \mathbb{R}^n \to \mathbb{R}$ *függvény lokális minimuma ha létezik az* x^* *pont olyan* $\mathcal{N}$ *környezete, melyben* $f(x^*) \leq f(x)$ *minden* $x \in \mathcal{N}$ *pontra.*

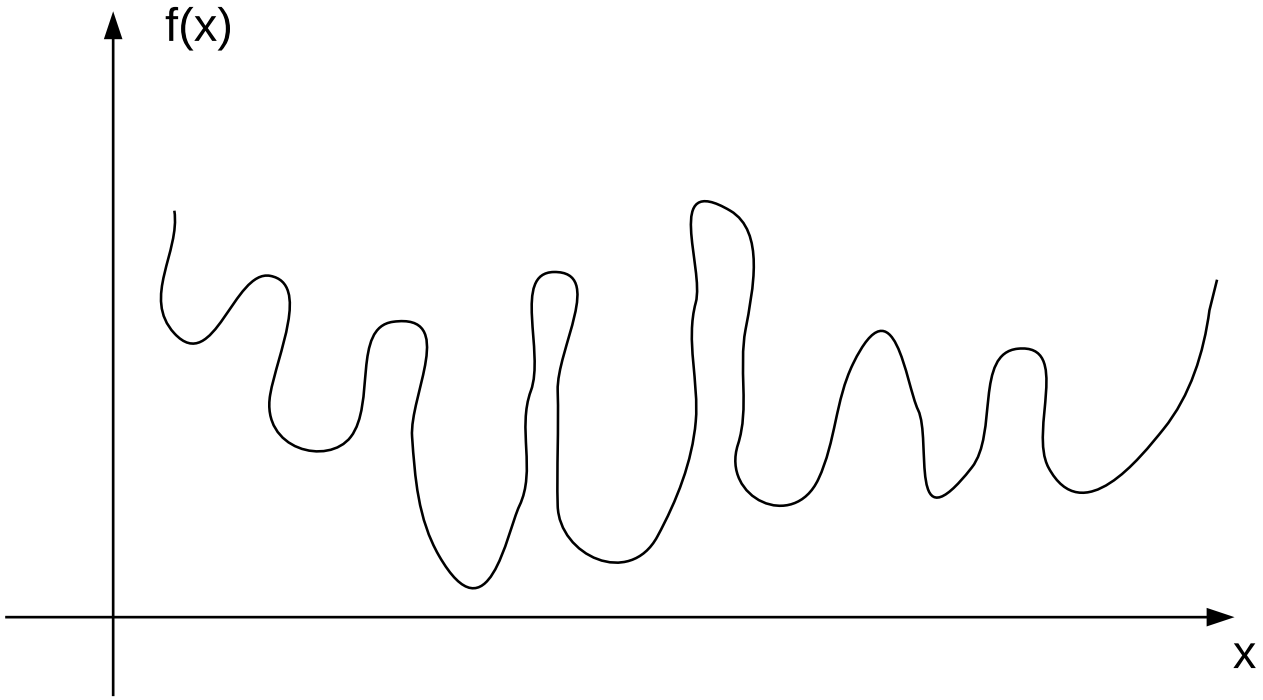
Definíció

Az $x^* \in \mathbb{R}^n$ *pont az* $f : \mathbb{R}^n \to \mathbb{R}$ *függvény szigorú lokális minimuma ha létezik az* x^* *pont olyan* $\mathcal{N}$ *környezete, melyben* $f(x^*) \leq f(x)$ *minden* $x \in \mathcal{N}, x \neq x^*$ *pontra.*

Mi a megoldása a feltétel nélküli optimalizálási feladatnak?



Mi a megoldása a feltétel nélküli optimalizálási feladatnak?



A feltétel nélküli maximalizálási feladat megoldása

- a maximalizálási feladatot át lehet vezetni minimalizálási feladatra;
- a globális, lokális és szigorú lokális maximumot analóg módon definiálják, mint a globális, lokális és szigorú lokális minimumot;

Direkt módszerek

Csak az f függvény értékeit használják (nem használnak deriváltakat). Az x^* minimumhelyet tartalmazó ún. befoglaló intervallumból indulnak ki és ezt szűkitik az eljárások során.

Direkt módszerek

Csak az f függvény értékeit használják (nem használnak deriváltakat). Az x^* minimumhelyet tartalmazó ún. befoglaló intervallumból indulnak ki és ezt szűkitik az eljárások során.

- a Nelder-Mead algoritmus;
- többirányú keresés;

Nelder-Mead algoritmus

Mire jó a Nelder-Mead algoritmus?

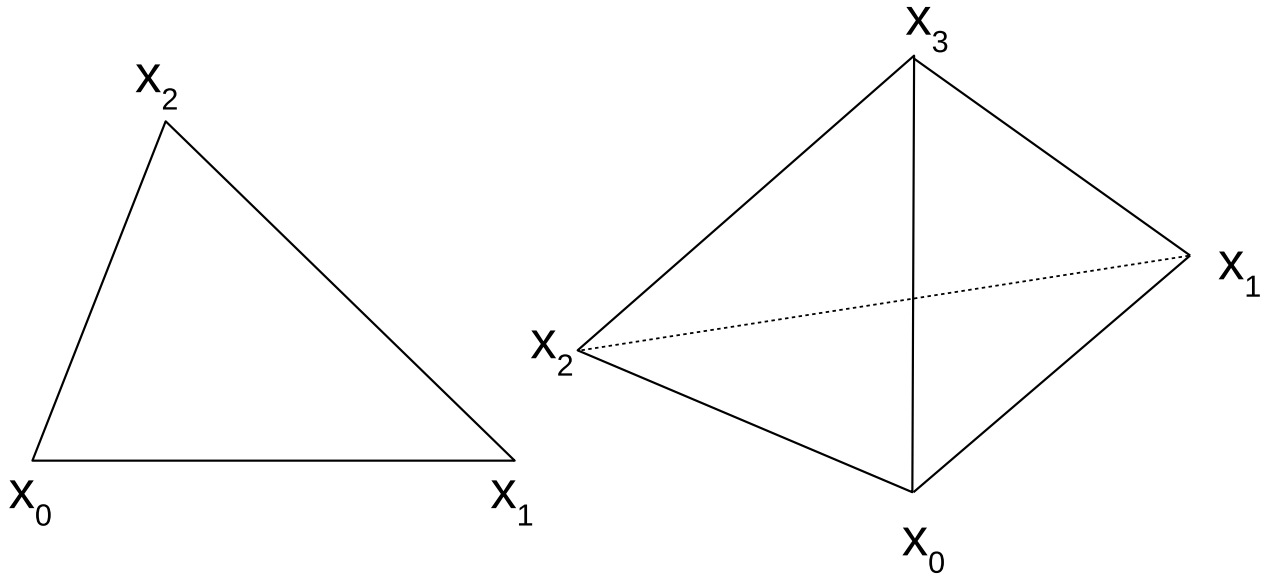
Az $f : \mathbb{R}^n \to \mathbb{R}$ folytonos függvény minimumámank meghatározása.

Nelder-Mead algoritmus

Mire jó a Nelder-Mead algoritmus?

Az $f : \mathbb{R}^n \to \mathbb{R}$ folytonos függvény minimumámank meghatározása.

A Nelder-Mead eljárás szimplex alapú. Például a szimplex az $\mathbb{R}^2$–ben egy háromszög, $\mathbb{R}^3$–ban egy tetraéder.



ábra. Két dimenziós szimplex (balra), három dimenziós szimplex (jobbra)

Nelder-Mead algoritmus

Egy lépés célja

A szimplex átalakítások segítségével csökkenteni a függvényértékeket a szimplex csúcsaiban.

Nelder-Mead algoritmus

Egy lépés célja

A szimplex átalakítások segítségével csökkenteni a függvényértékeket a szimplex csúcsaiban.

A Nelder-Mead algoritmus csak egy vagy maximum két függvény kiértékelést végez minden egyes lépésben, amíg sok más direkt kereső módszer n vagy még több függvény kiértékelést használ.

Nelder-Mead algoritmus

Általános algoritmus

1. lépés Konstruáljuk meg a kezdő S szimplexet.
2. lépés Ismételjük az alábbi lépéseket, amíg a leállási feltételek nem teljesülnek:
   a) Számoljuk ki a leállási feltételeket;
   b) Ha a leállási feltételek nem teljesülnek, akkor alakítsuk át az aktuális szimplexet a következő transzformációkkal:
      - tükrözés (reflection);
      - összehúzás (contraction);
      - tágítás/nyújtás (expansion);
      - zsugorítás (shrinkage).
3. lépés Ha a leállási feltételek teljesülnek, akkor az aktuális szimplex legjobb csúcsával tér vissza és a hozzákapcsolódó függvényértékkel.

Nelder-Mead algoritmus

A Nelder-Mead módszer megvalósításra sok különféle lehetőség adódik. A fő különbség a változatos kivitelezésekben a kezdő szimplex megkonstruálásában, és az iterációs folyamat végére adott leállási feltételek meghatározásában rejlik.

Nelder-Mead algoritmus

A Nelder-Mead módszer megvalósításra sok különféle lehetőség adódik. A fő különbség a változatos kivitelezésekben a kezdő szimplex megkonstruálásában, és az iterációs folyamat végére adott leállási feltételek meghatározásában rejlik.

Kezdő szimplex

Generáljunk $n+1$ csúcsot $x_0, x_1, \ldots, x_n$ egy adott bemenő pont x_{in} köré. Általában $x_0 = x_{in}$. A szimplex megmaradó n csúcsát a következő módon generáljuk: $x_j = x_0 + h_j e_j, j = 1, \ldots, n$, ahol h_j egy lépés méret az e_j egység vektor irányába.

Nelder-Mead algoritmus

A Nelder-Mead algoritmus egy iterációs lépése a következő három lépésből áll:

1. lépés: Sorba rendezés

Határozzunk meg három indexet az aktuális szimplexre vonatkozva:

h legrosszabb csúcs;

s második legrosszabb csúcs;

l a legjobb csúcs

Jelölés: $f_i = f(x_i)$, $f_h = \max_j f_j$, $f_s = \max_{j \neq h} f_j$, $f_l = \min_{j \neq h} f_j$.
Néhány megvalósításban az S csúcsai sorba vannak rendezve a függvényértékekre vonatkozóan: $f_0 \leq f_1 \leq \cdots \leq f_n$. Ekkor $l = 0$, $s = n - 1$, és $h = n$.

Nelder-Mead algoritmus

A Nelder-Mead algoritmus egy iterációs lépése a következő három lépésből áll:

2. lépés: Középpontosítás (centroid kiszámítása)

Kiszámítjuk a legjobb oldal centroidját c. Ez az egyetlen, ami szemben helyezkedik a legrosszabb csúccsal x_h:

$$c = \frac{1}{n} \sum_{j \neq h} x_j.$$

Nelder-Mead algoritmus

A Nelder-Mead algoritmus egy iterációs lépése a következő három lépésből áll:

3. lépés: Transzformáció

Kiszámítjuk az új szimplexet az aktuálisból. Először is próbáljuk csak a legrosszabb x_h csúcsot visszahívni egy jobb ponttal tükrözés, tágítás vagy összehúzás segítségével, figyelemmel a legjobb oldalra. Az összes tesztpont egy vonalon fekszik definiálva x_h és c által, és legfeljebb kettőt közülük kiszámítunk egy iterációs lépésen belül. Ha ez sikert ér el, akkor az elfogadott pont válik a szimplex új csúcsává. Ha sikertelen, akkor zsugorítjuk a szimplexet a legjobb csúcs x_j felé. Ebben az esetben n új csúcs lett kiszámítva.

Nelder-Mead algoritmus

Szimplex transzformációk

A szimplex transzformációkat a Nelder-Mead módszerben négy paraméter kontrollálja:

- tükrözés (reflection): α;
- összehúzás (contraction): β;
- tágítás/nyújtás (expansion): γ;
- zsugorítás (shrinkage): δ;

$$\alpha > 0,\ 0 < \beta < 1,\ \gamma > \max\{1, \alpha\},\ 0 < \delta < 1.$$

A standard értékek a következők:

$$\alpha = 1,\ \beta = \frac{1}{2},\ \gamma = 2,\ \delta = \frac{1}{2}.$$

Nelder-Mead algoritmus

Tükrözés (reflectioin)

Számítsuk ki a tükrözési pontot:

$$x_r := c + \alpha(c - x_h) \text{ és } f_r := f(x_r).$$

Ha $f_l \leq f_r < f_s$, elfogadjuk x_r-t és befejezzük az iterációt.

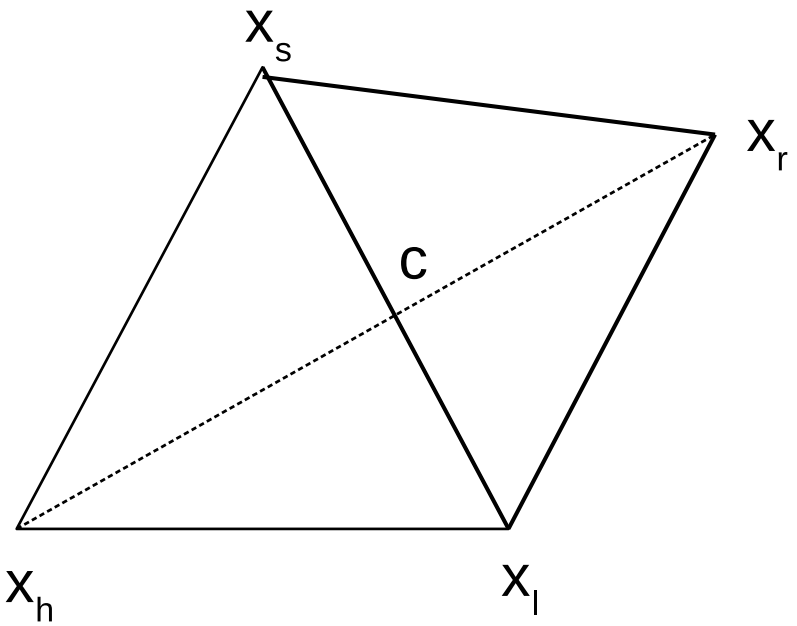
Nelder-Mead algoritmus

Tükrözés (reflectioin)

Számítsuk ki a tükrözési pontot:

$$x_r := c + \alpha(c - x_h) \text{ és } f_r := f(x_r).$$

Ha $f_l \leq f_r < f_s$, elfogadjuk x_r-t és befejezzük az iterációt.



Nelder-Mead algoritmus

Tágítás/nyújtás (expansion)

Ha $f_r < f_l$, számítsuk ki a nyújtási pontot

$$x_e := c + (x_r - c) \text{ és } f_e := f(x_e).$$

Ha $f_e < f_r$, elfogadjuk x_e-t és befejezzük az iterációt. Másképpen ($f_e \geq f_r$), elfogadjuk x_r-t és befejezzük az iterációt.

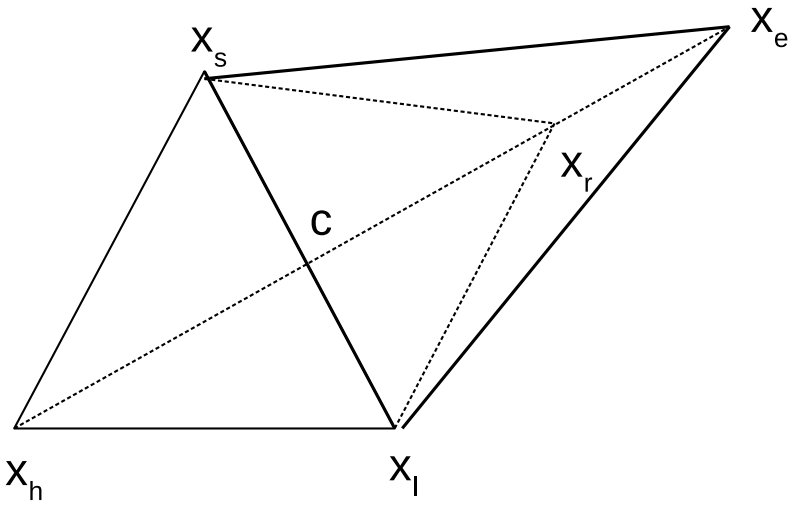
Nelder-Mead algoritmus

Tágítás/nyújtás (expansion)

Ha $f_r < f_l$, számítsuk ki a nyújtási pontot

$$x_e := c + (x_r - c) \text{ és } f_e := f(x_e).$$

Ha $f_e < f_r$, elfogadjuk x_e-t és befejezzük az iterációt. Másképpen ($f_e \geq f_r$), elfogadjuk x_r-t és befejezzük az iterációt.



Nelder-Mead algoritmus

Összehúzás (contract)

Ha $f_r \geq f_s$, kiszámítjuk az összehúzási pontot x_c, amire a két legjobb pontot használjuk x_h-t és x_r-t.

Kifelé: Ha $f_s \leq f_r < f_h$, kiszámítjuk

$$xc := c + \beta(x_r - c) \text{ és } f_c := f(x_c).$$

Ha $f_c \leq f_r$, elfogadjuk x_c-t és befejezzük az iterációt. Egyébként, végrehajtunk egy zsugorítási transzformációt.

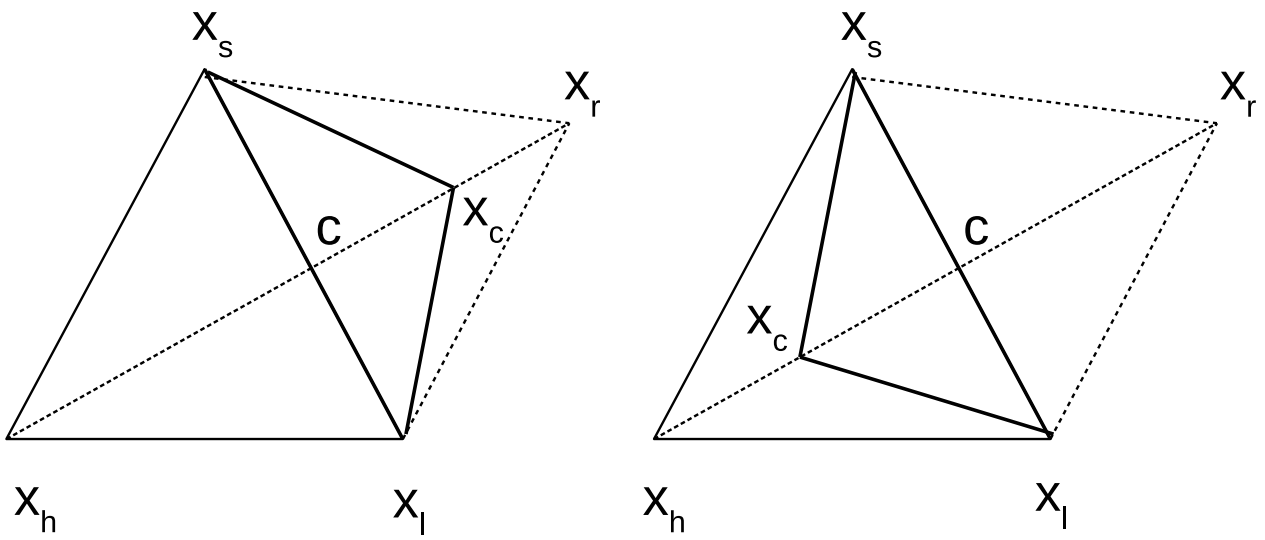
Befelé: Ha $f_r \geq f_h$, kiszámítjuk

$$xc := c + \beta(x_h - c) \text{ és } f_c := f(x_c).$$

Ha $f_c < f_h$, elfogadjuk x_c-t és befejezzük az iterációt. Egyébként, végrehajtunk egy zsugorítási transzformációt.

Nelder-Mead algoritmus

Összehúzás (contract)



Nelder-Mead algoritmus

Zsugorítás (shrinkage)

Kiszámítjuk n új csúcsra

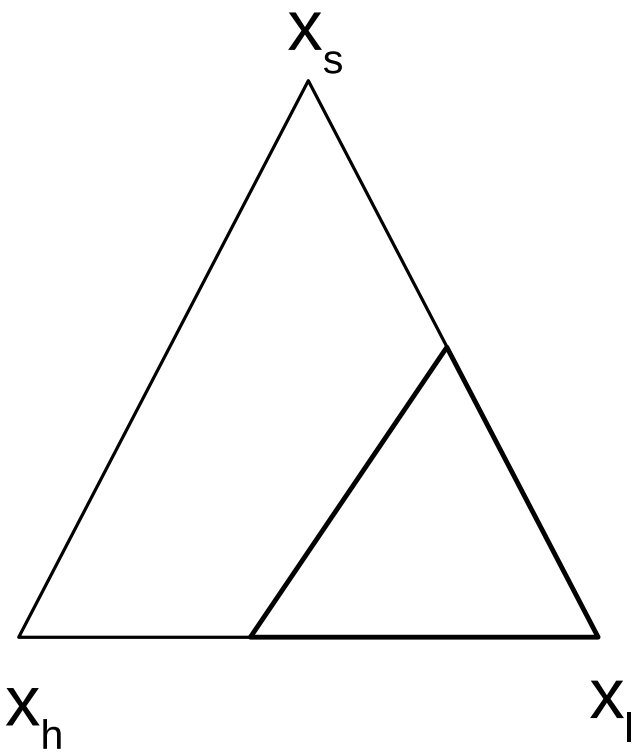
$$x_j := x_l + \delta(x_j - x_l) \text{ és } f_j := f(x_j), j = 0, \ldots, n \text{ és } j \neq l.$$

Nelder-Mead algoritmus

Zsugorítás (shrinkage)

Kiszámítjuk n új csúcsra

$$x_j := x_l + \delta(x_j - x_l) \text{ és } f_j := f(x_j), j = 0, \ldots, n \text{ és } j \neq l.$$



Nelder-Mead algoritmus

Nelder-Mead algoritmus akcióban

Nelder-Mead akcióban

Nelder-Mead algoritmus

Nelder-Mead algoritmus akcióban

Nelder-Mead akcióban

Nelder-Mead algoritmus Matlab-ban

Nelder-Mead algoritmus Matlabban

Nelder-Mead algoritmus

1. Feladat

Matlab-ban határozza meg az $f(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 8)^4$ *függvény minimumát.*

Nelder-Mead algoritmus

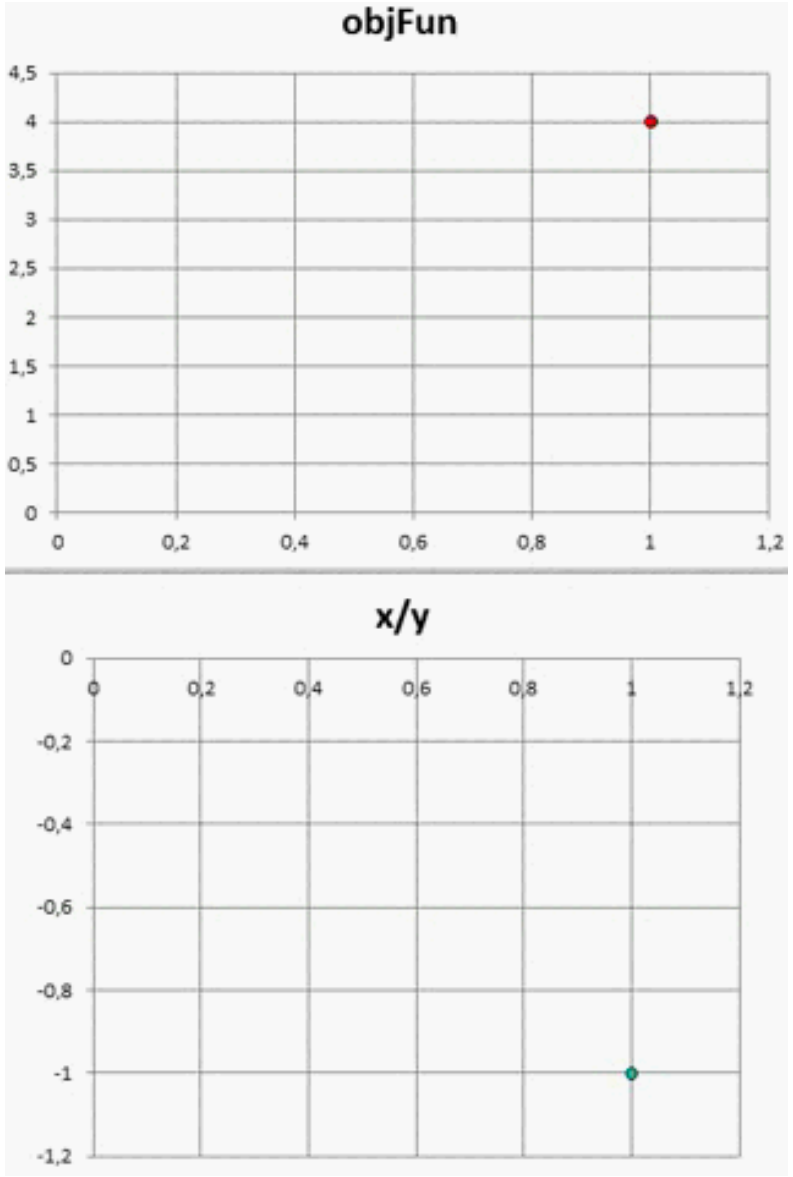
1. Feladat

Matlab-ban határozza meg az $f(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 8)^4$ *függvény minimumát.*

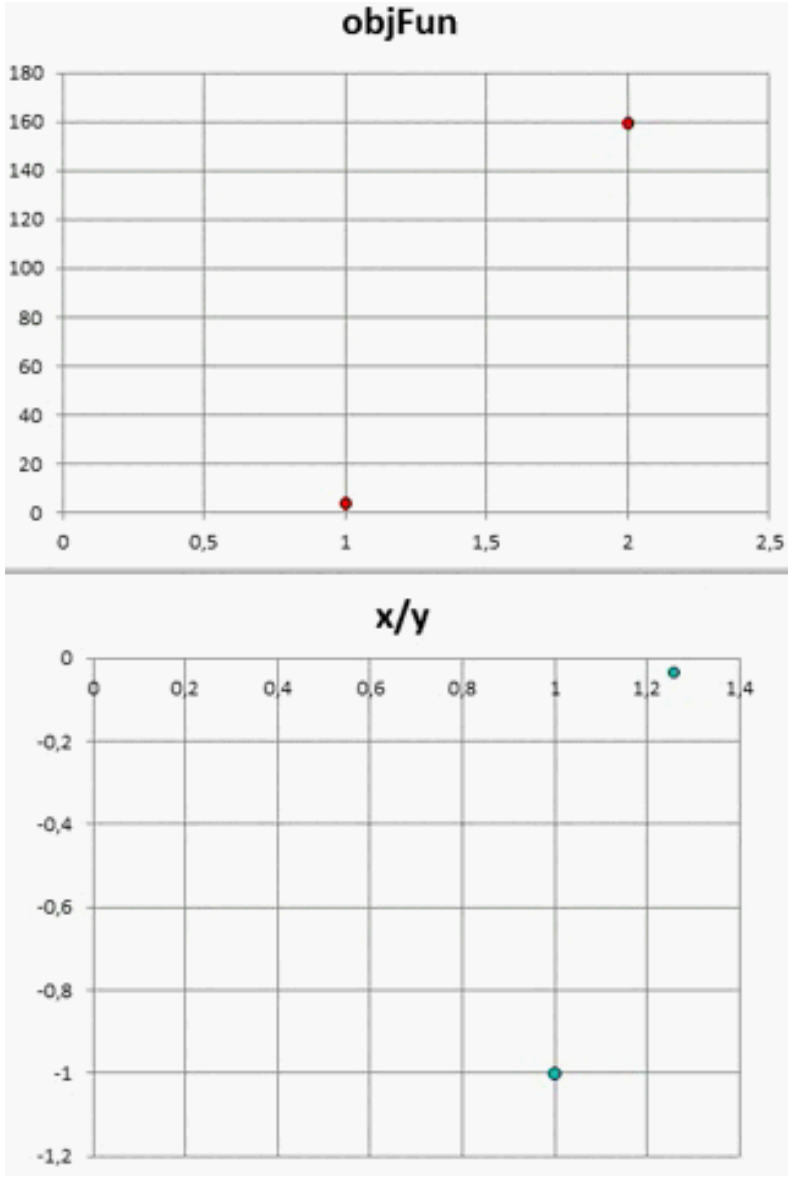
2. Feladat

Matlab-ban határozza meg az $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$ *Rosenbrock függvény minimumát úgy, hogy a függvényt m-file alakjában definiálja.*

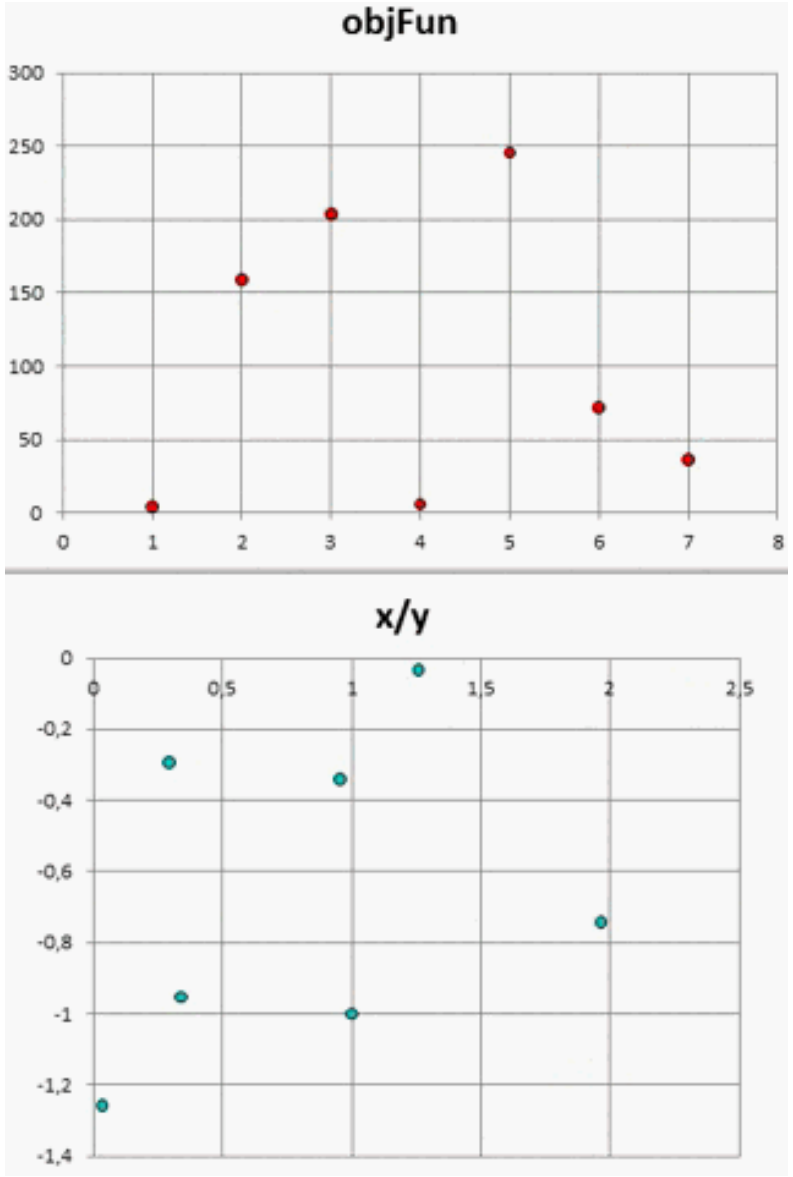
Nelder-Mead algoritmus



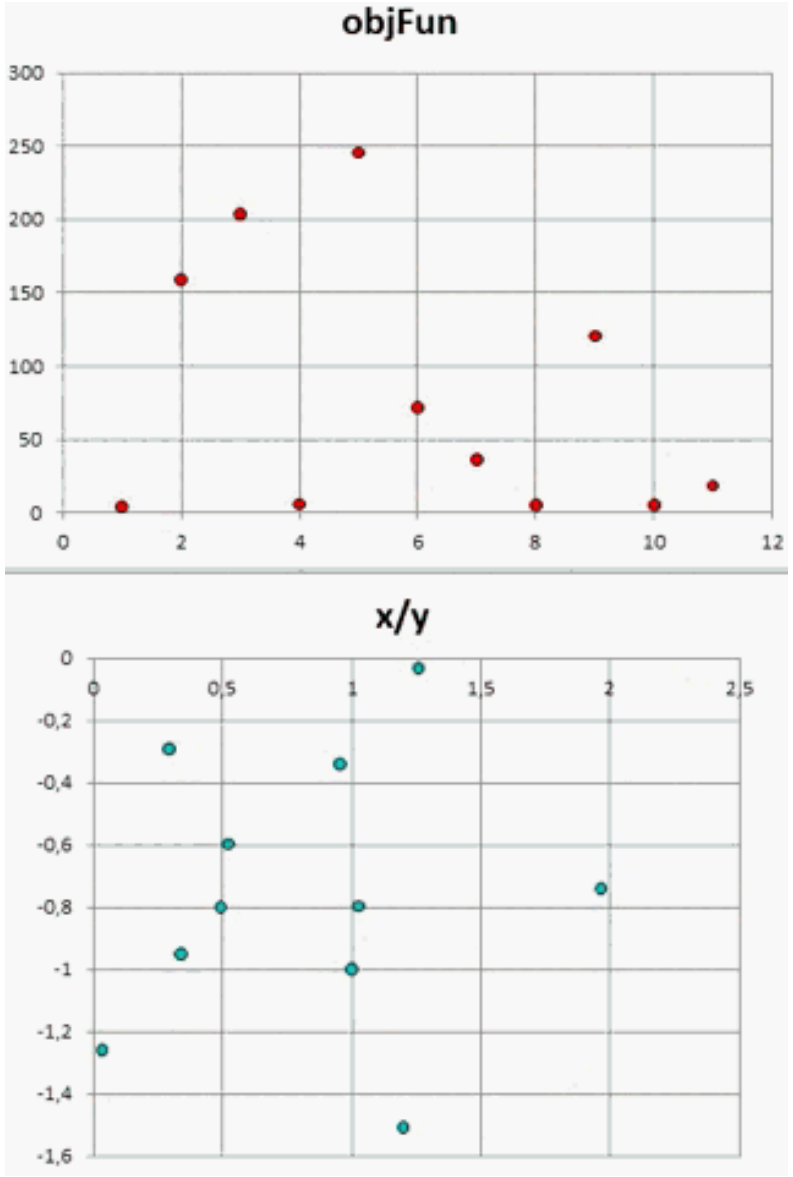
Nelder-Mead algoritmus



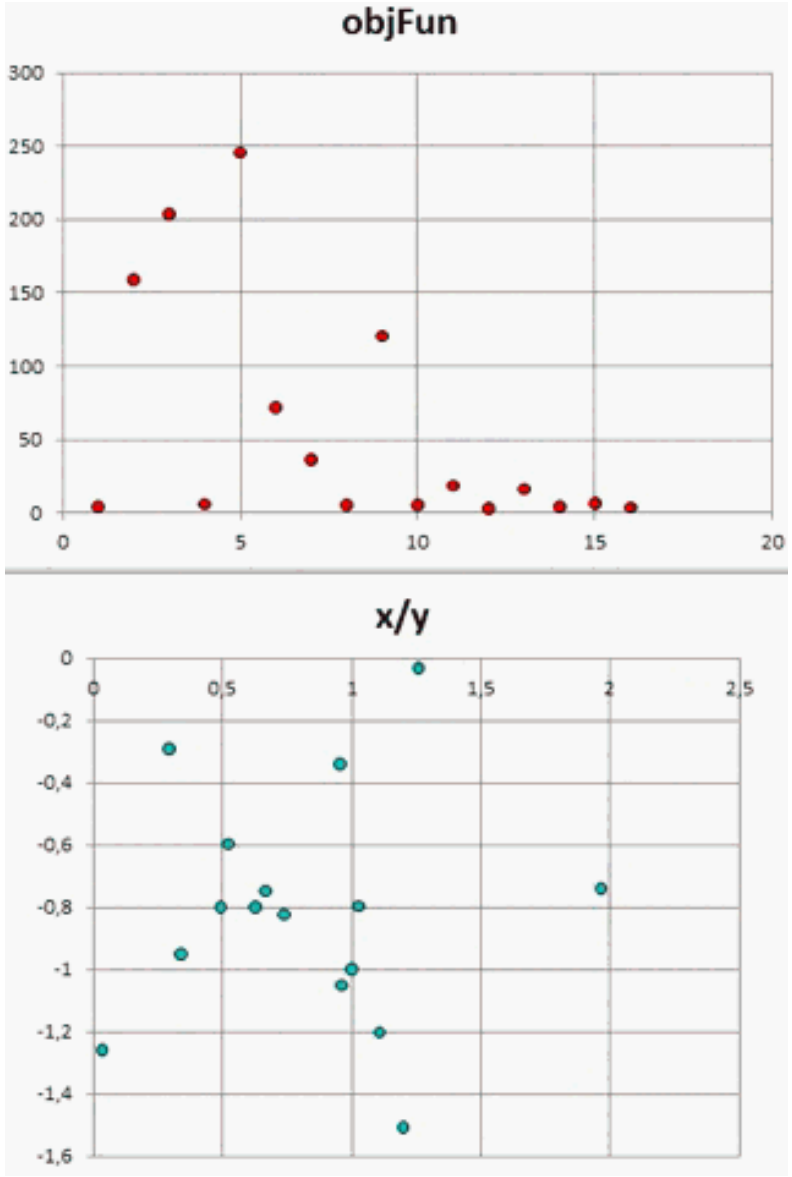
Nelder-Mead algoritmus



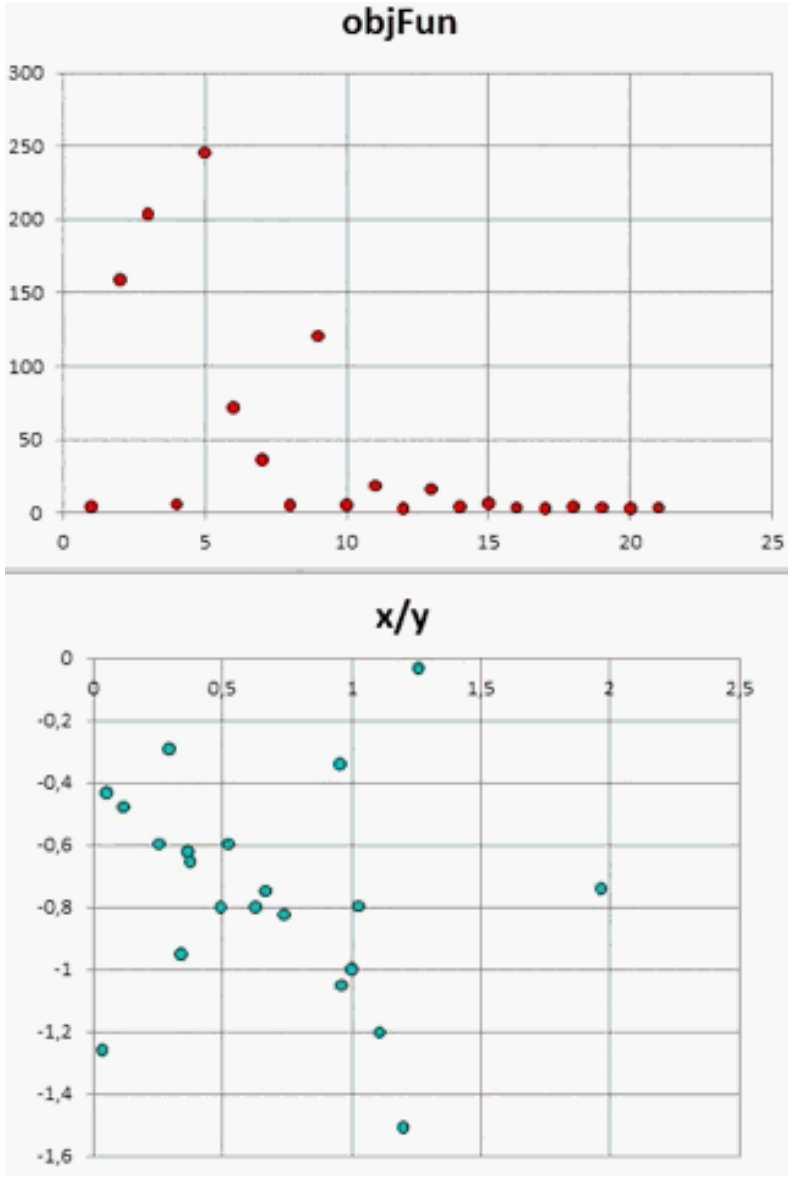
Nelder-Mead algoritmus



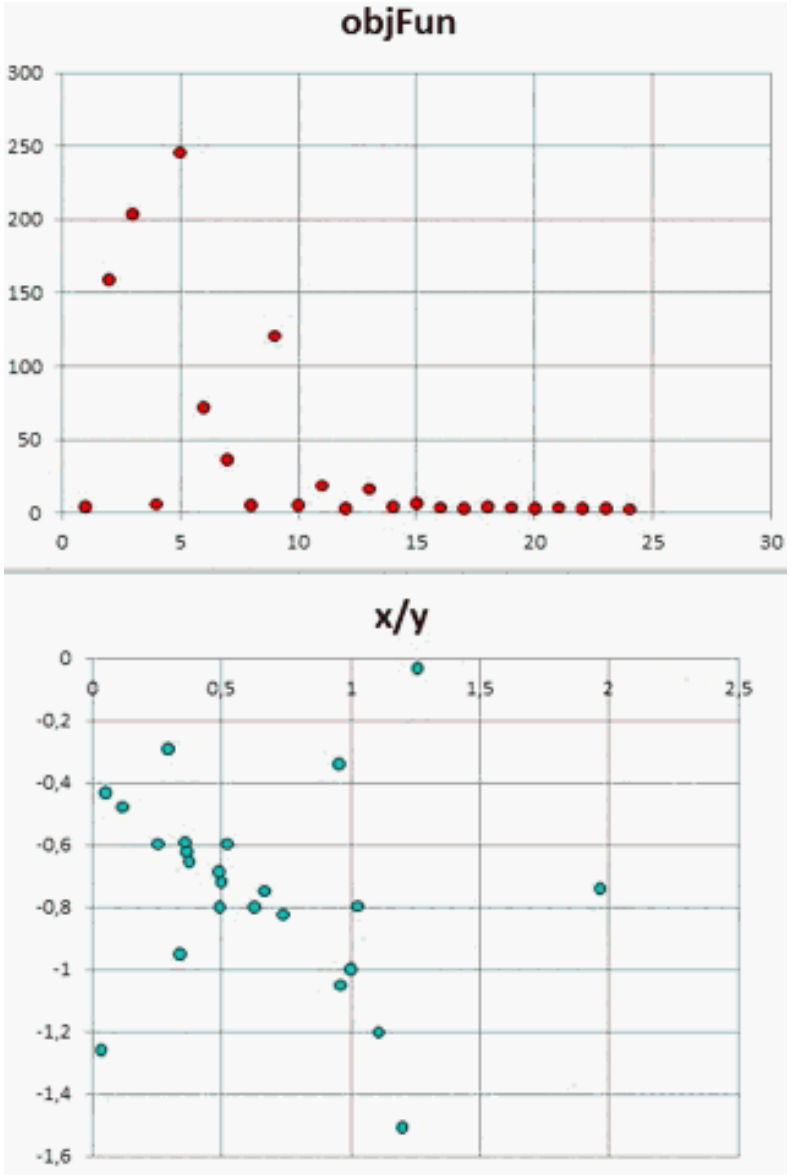
Nelder-Mead algoritmus



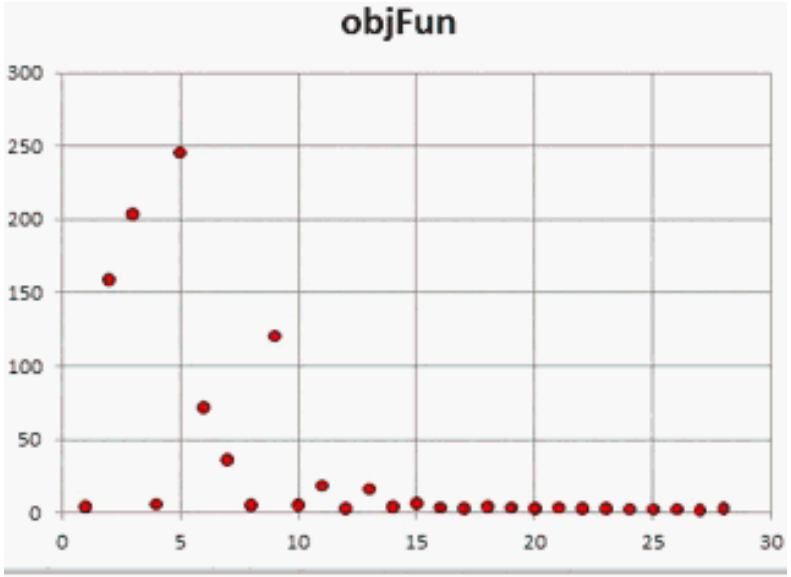
Nelder-Mead algoritmus

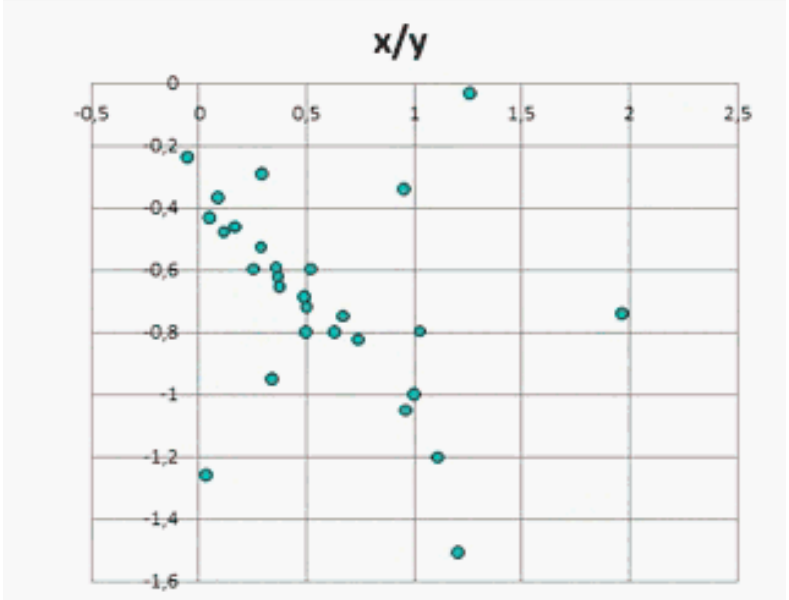


Nelder-Mead algoritmus

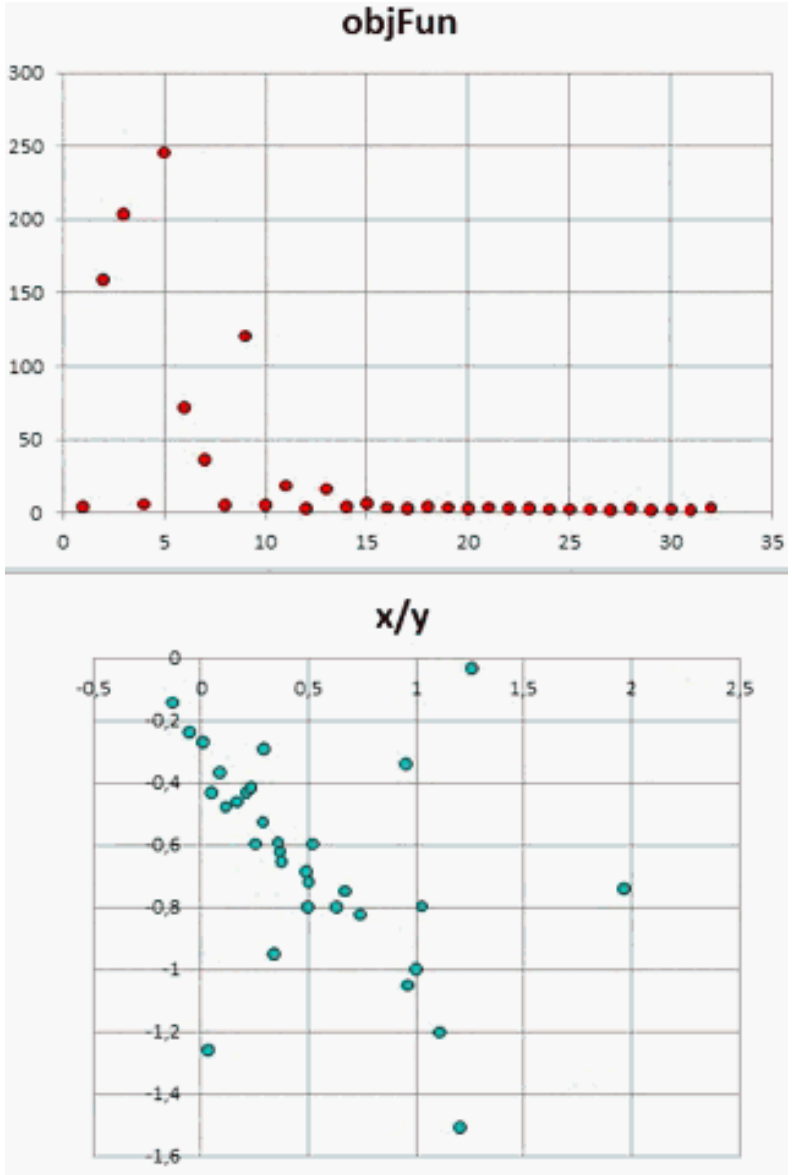


Nelder-Mead algoritmus

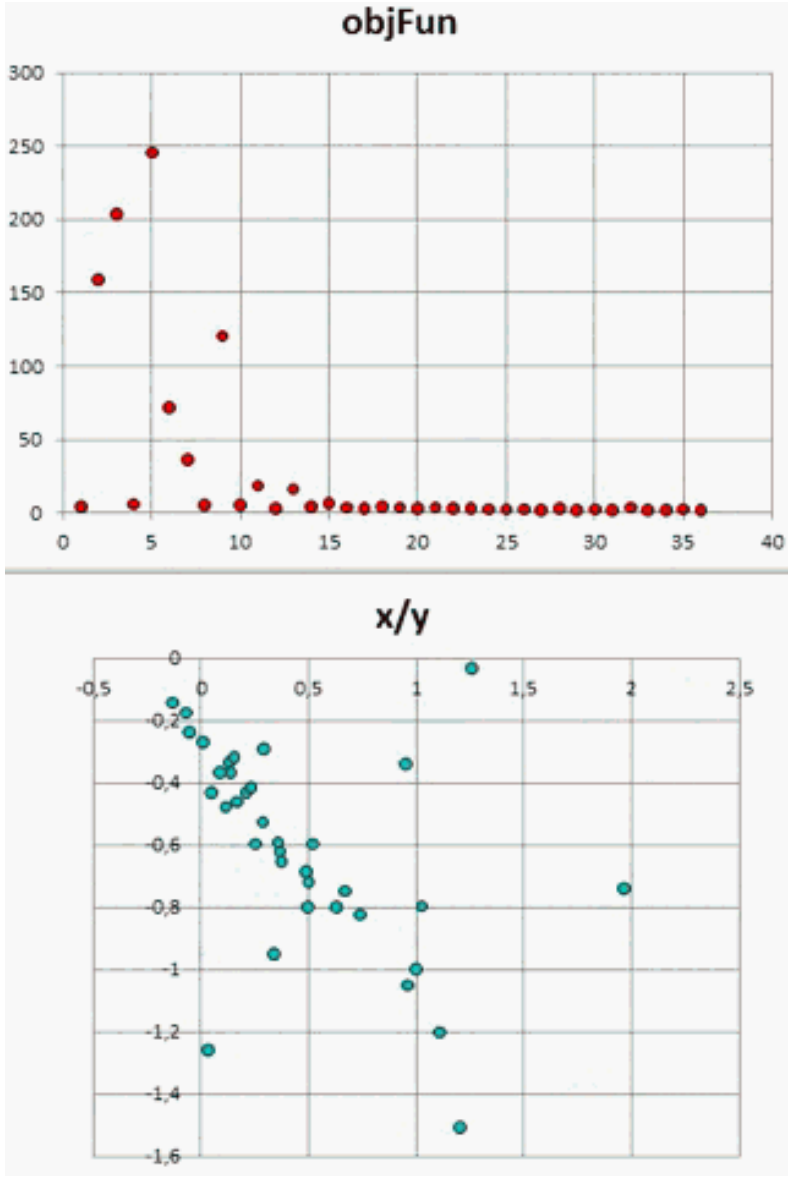




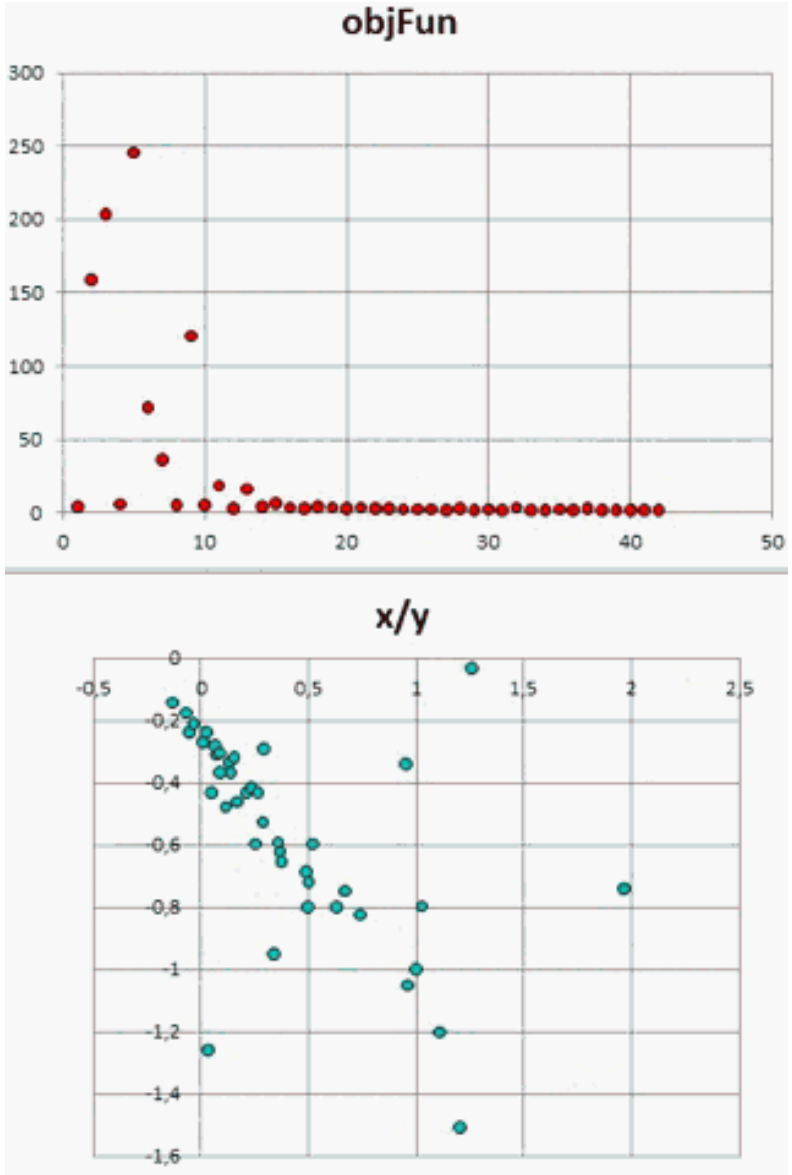
Nelder-Mead algoritmus



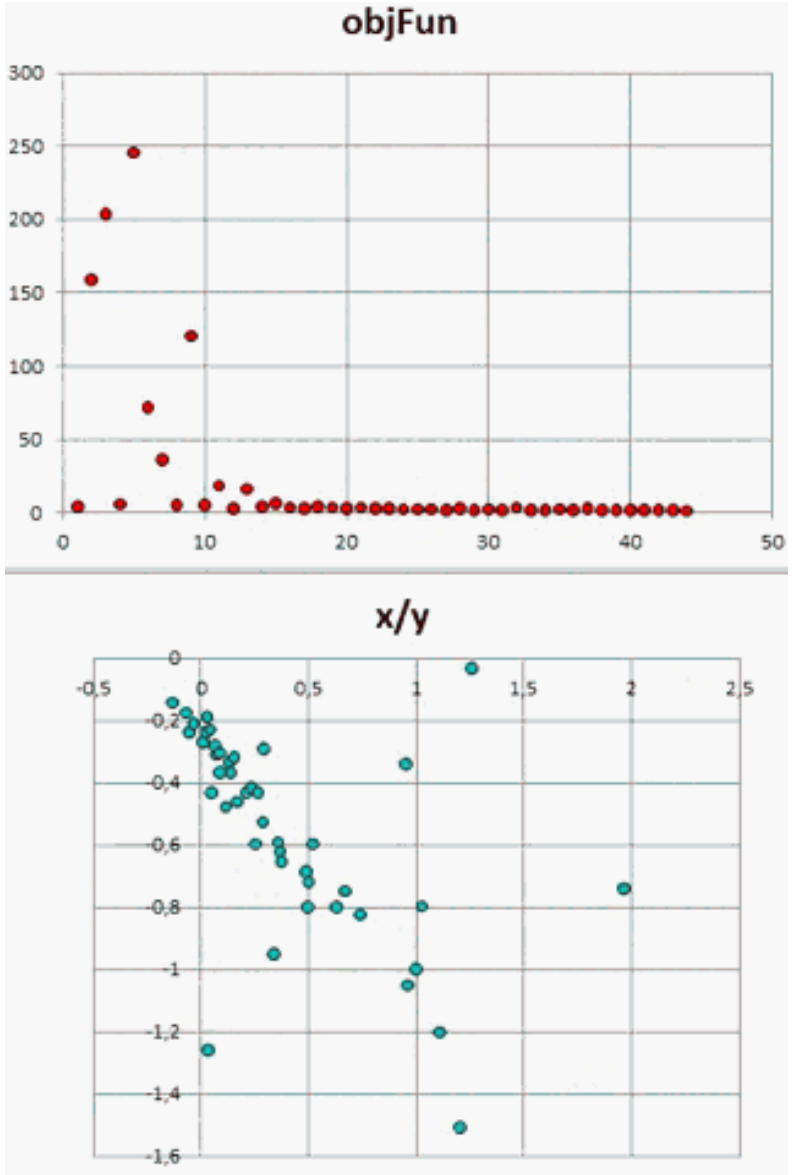
Nelder-Mead algoritmus



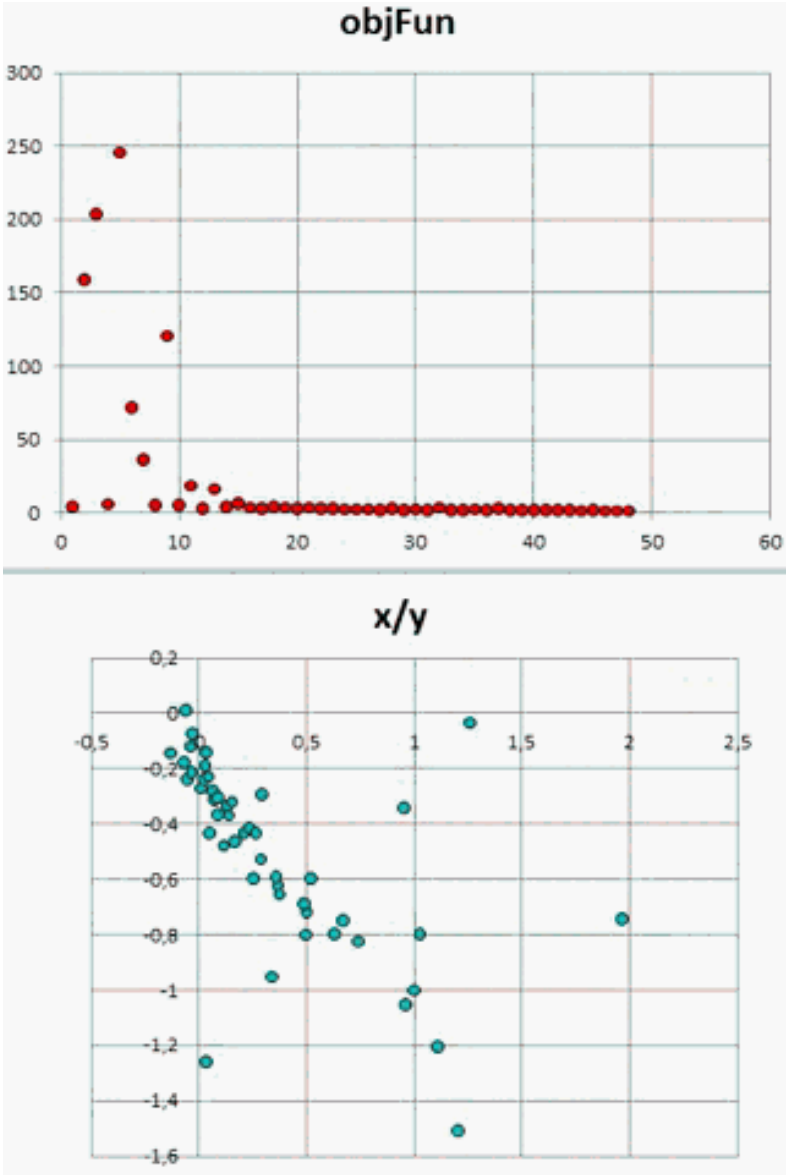
Nelder-Mead algoritmus



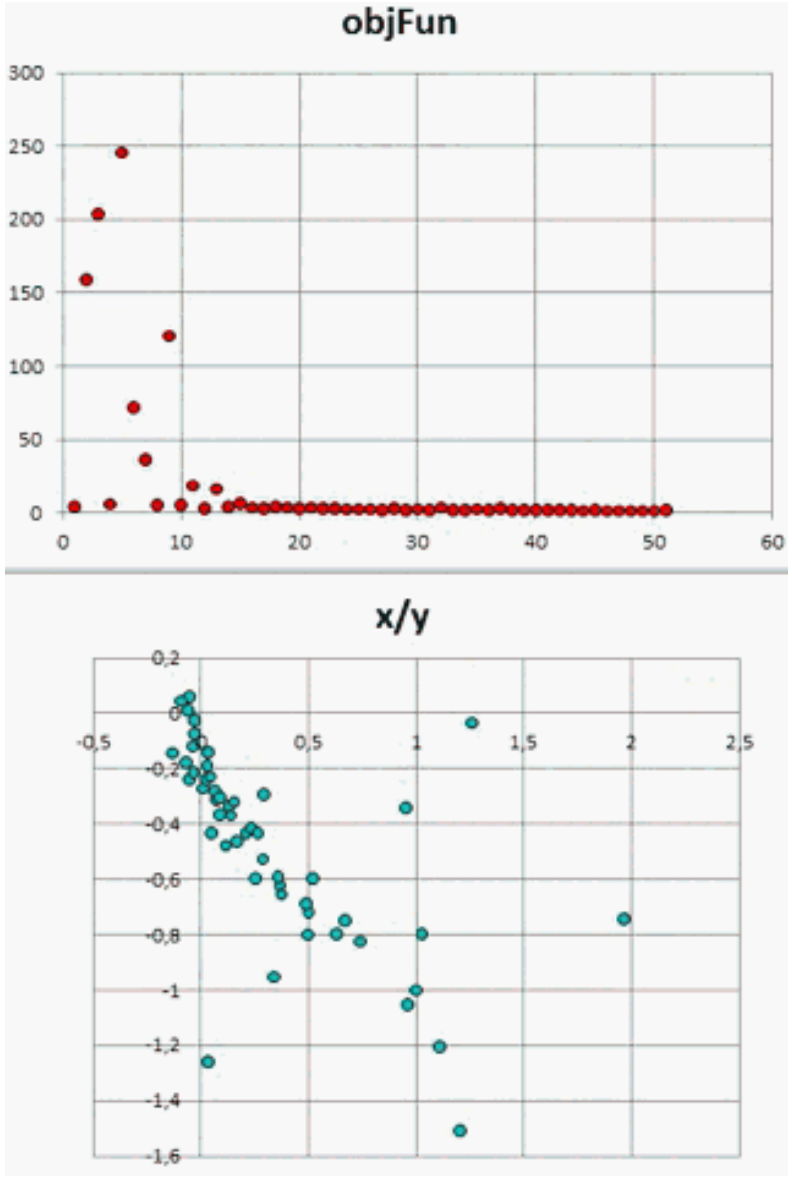
Nelder-Mead algoritmus



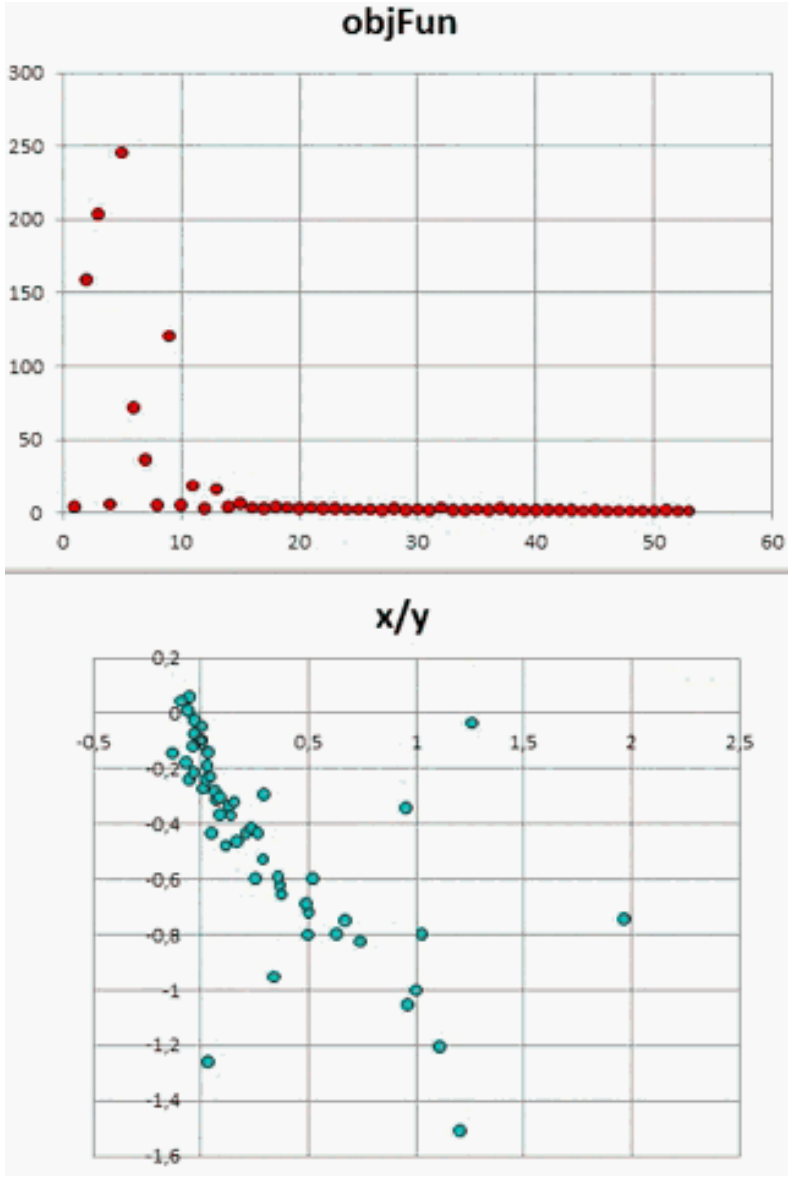
Nelder-Mead algoritmus



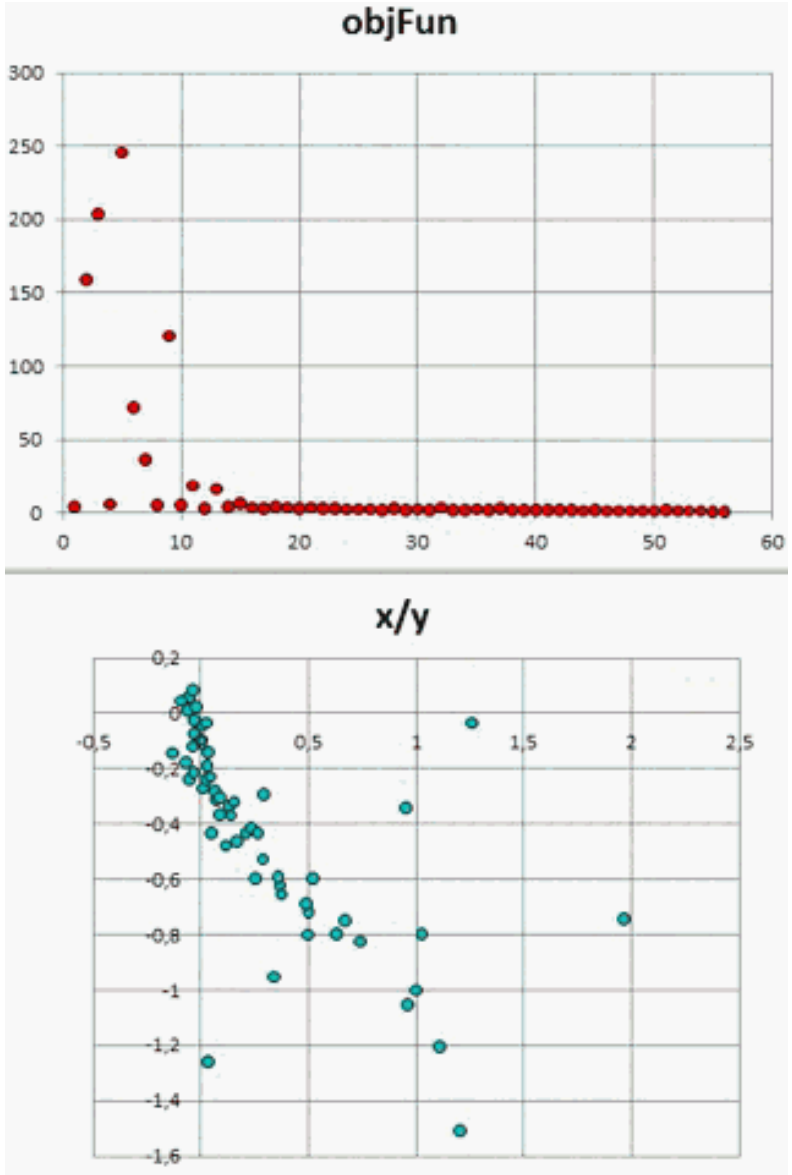
Nelder-Mead algoritmus



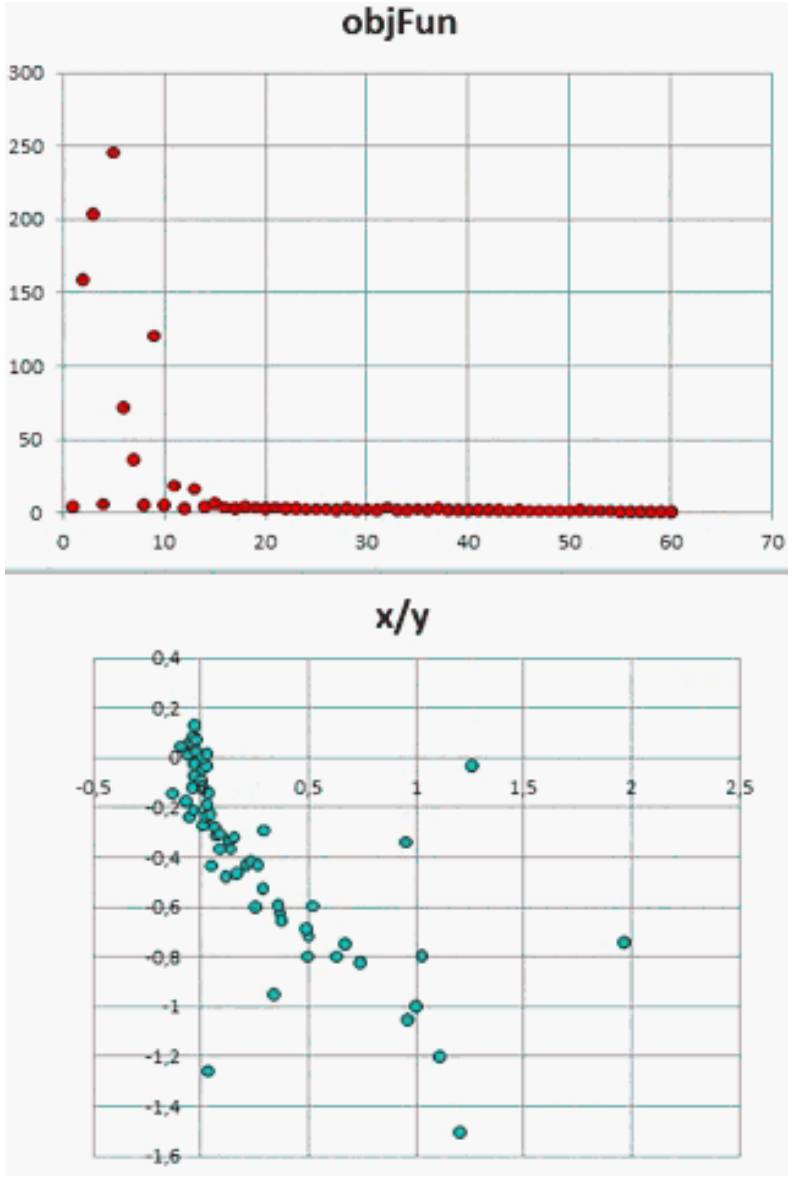
Nelder-Mead algoritmus



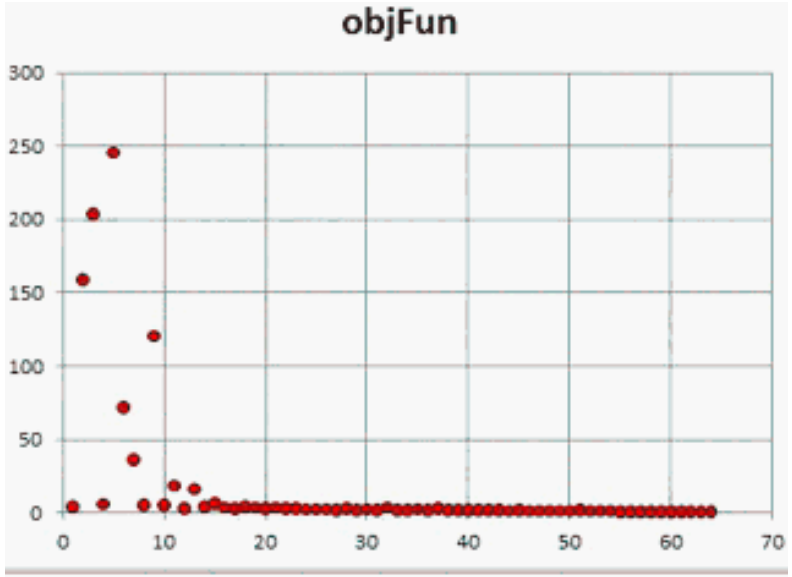
Nelder-Mead algoritmus

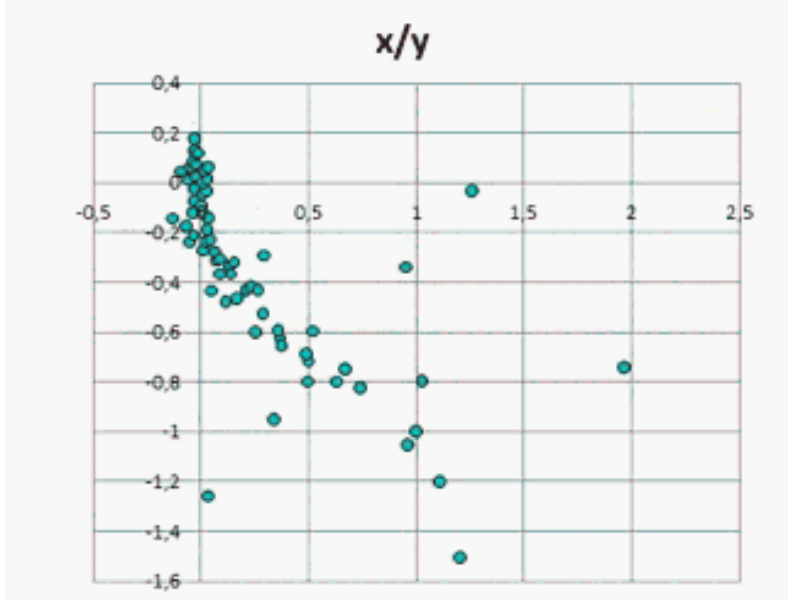


Nelder-Mead algoritmus

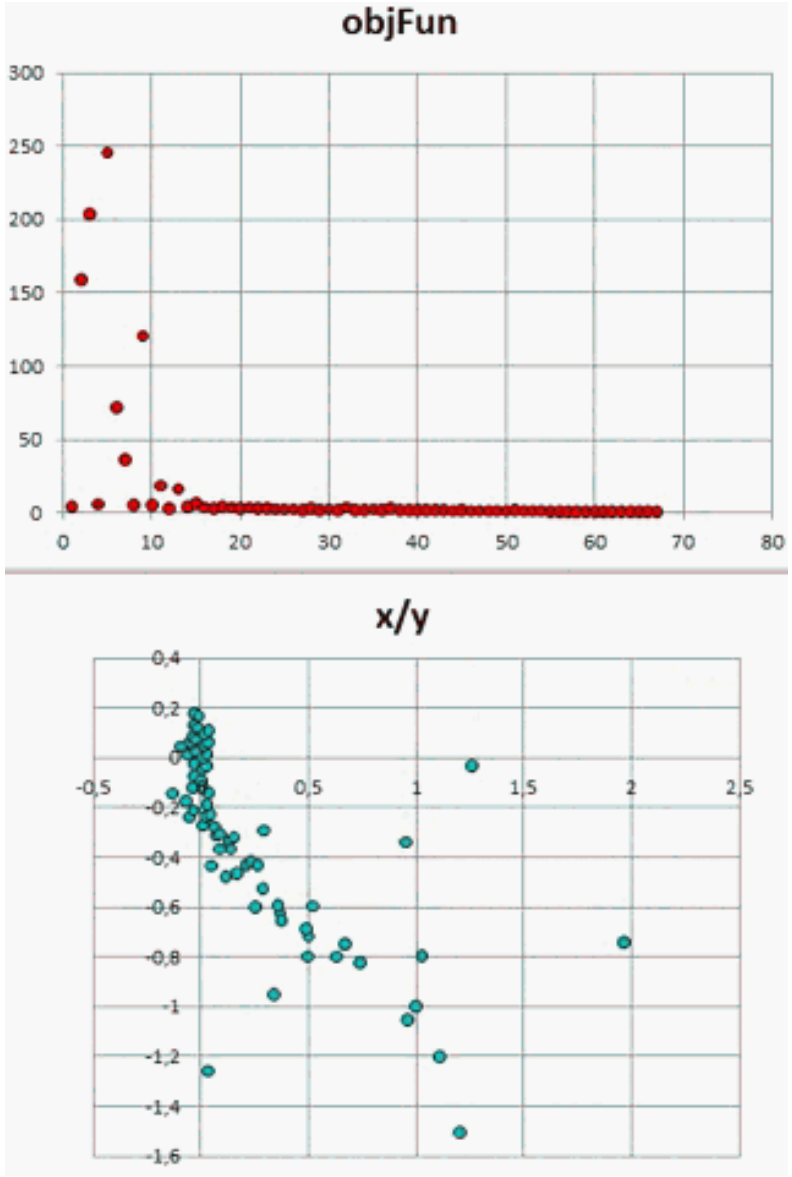


Nelder-Mead algoritmus

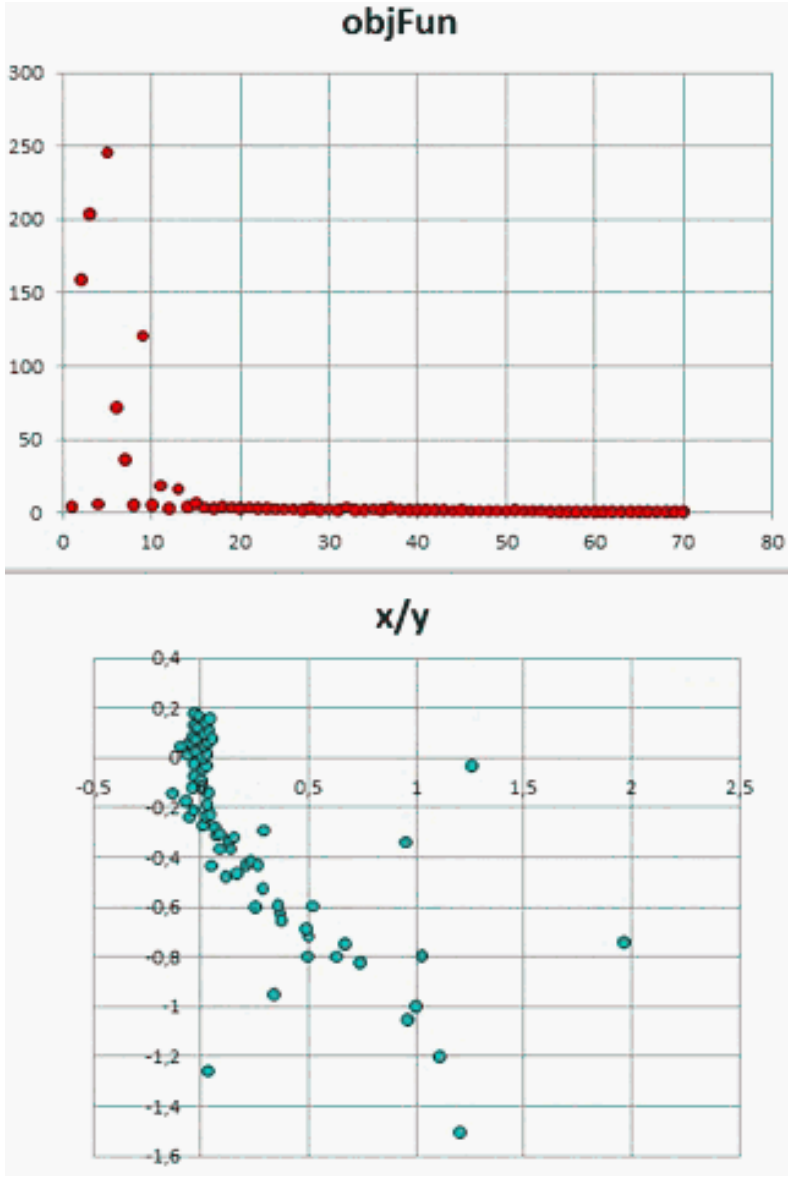




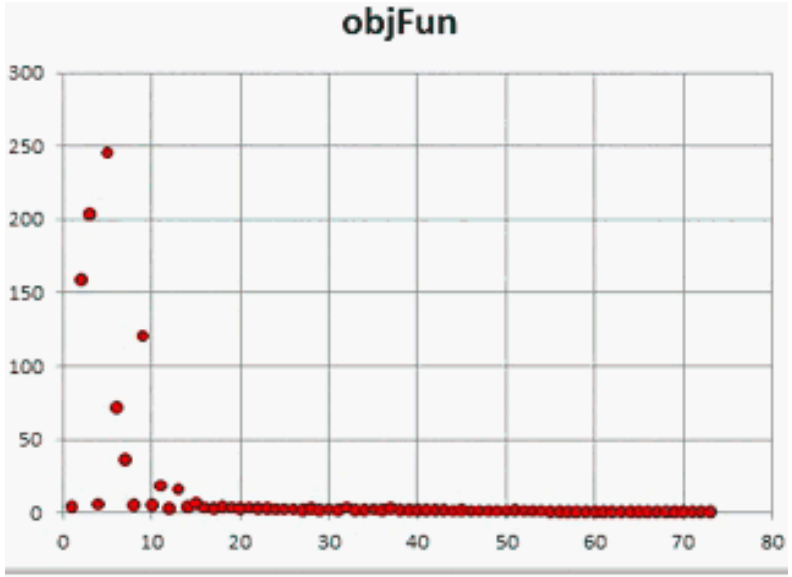
Nelder-Mead algoritmus

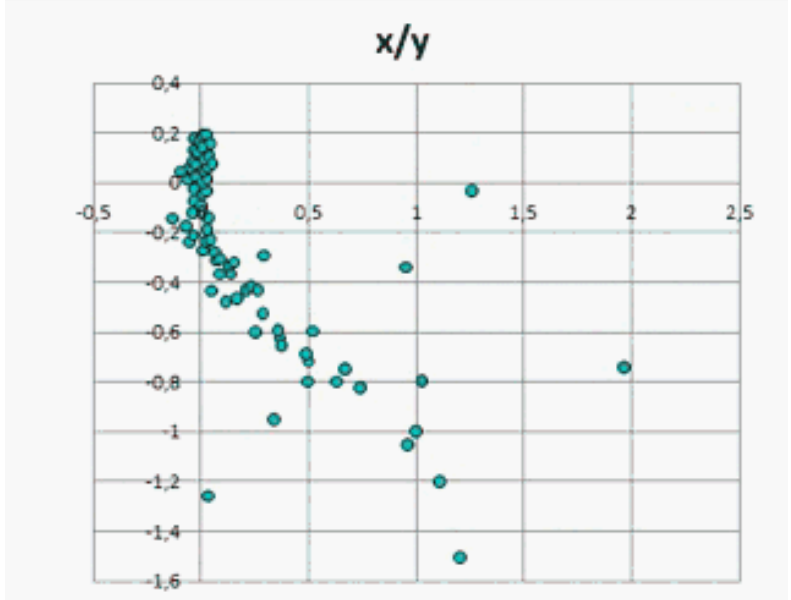


Nelder-Mead algoritmus

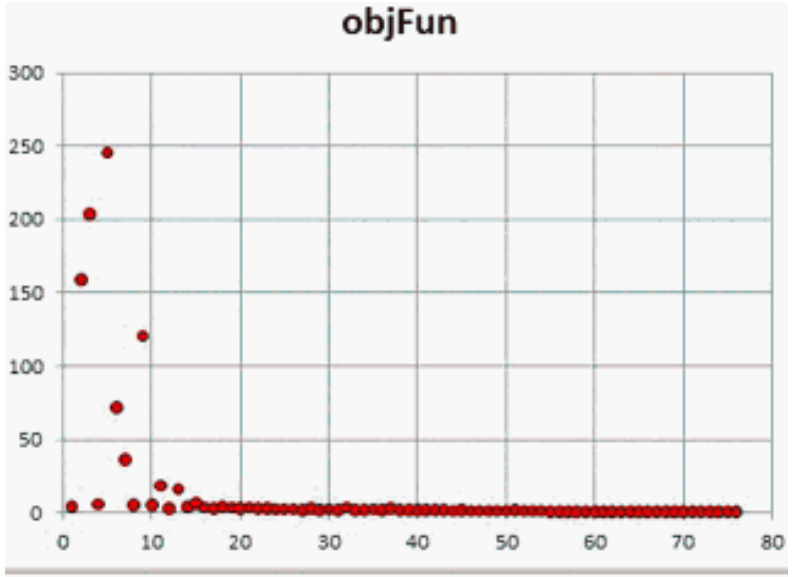


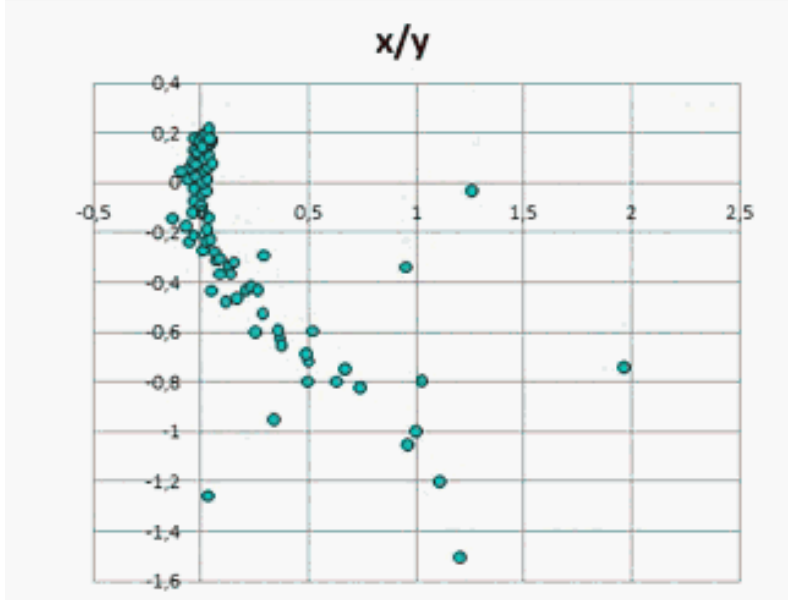
Nelder-Mead algoritmus



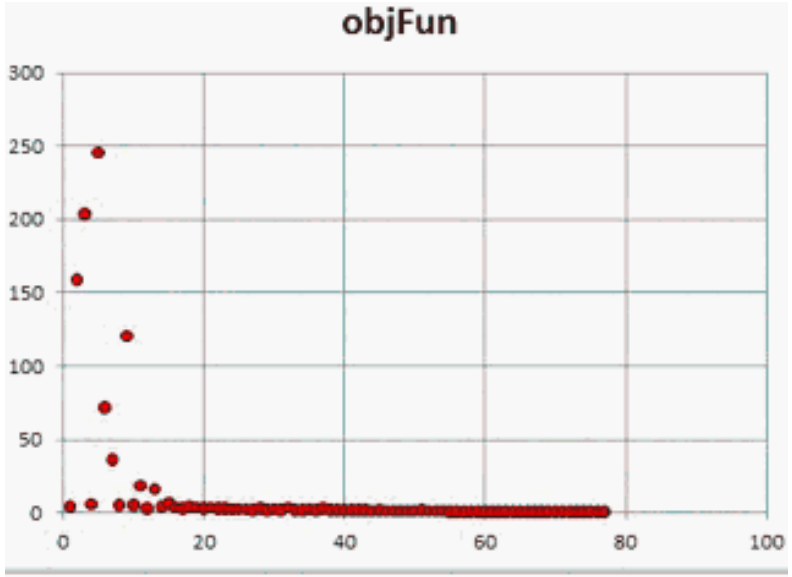


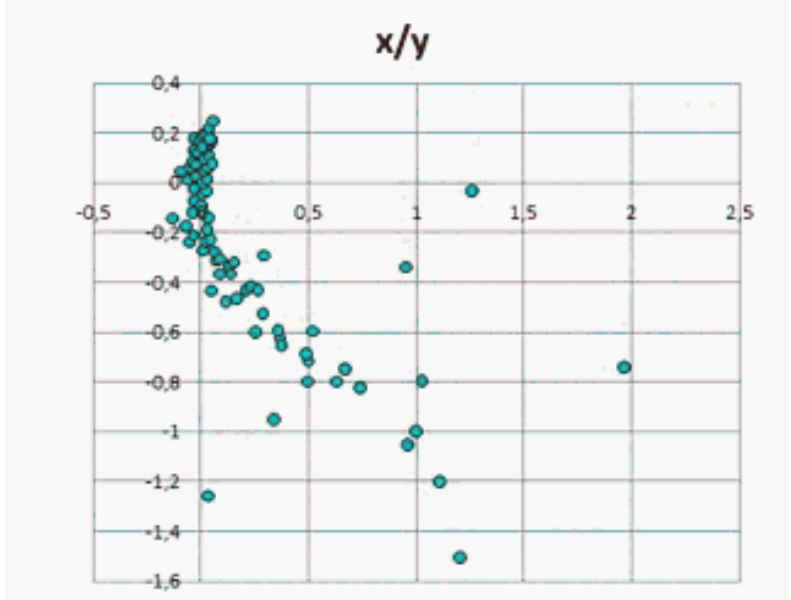
Nelder-Mead algoritmus



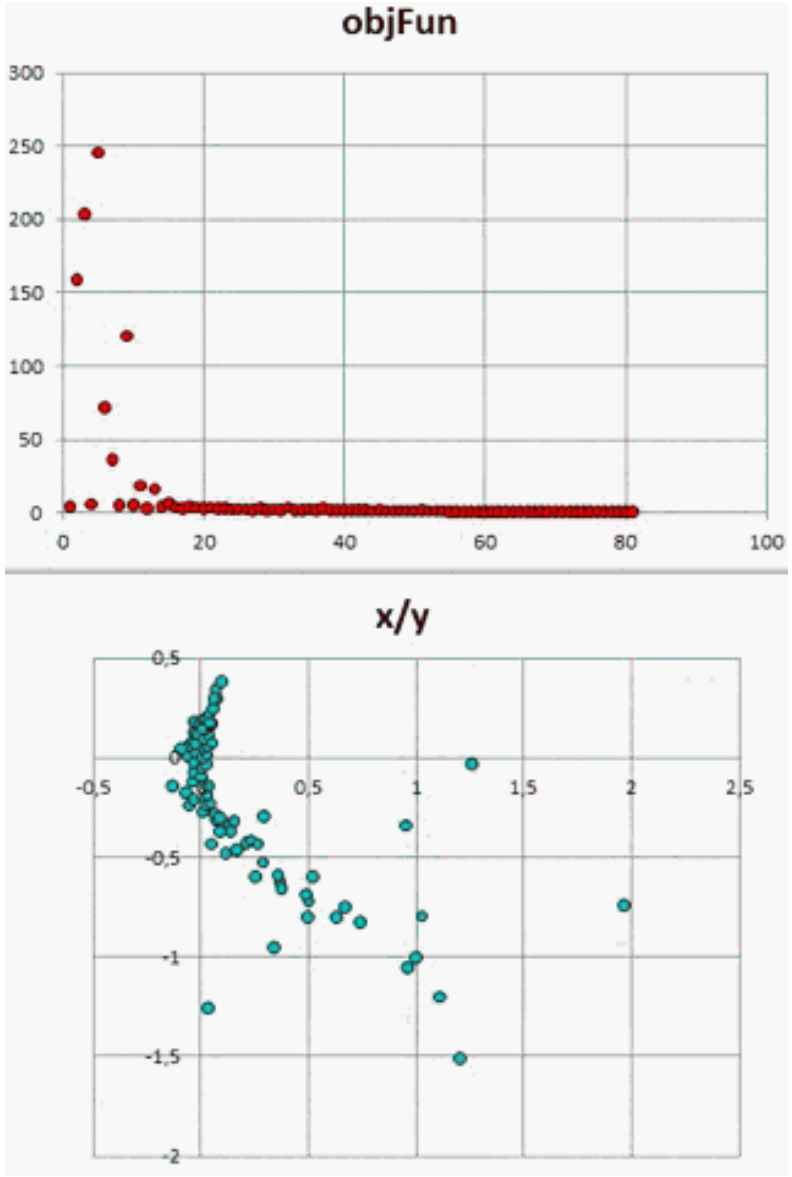


Nelder-Mead algoritmus

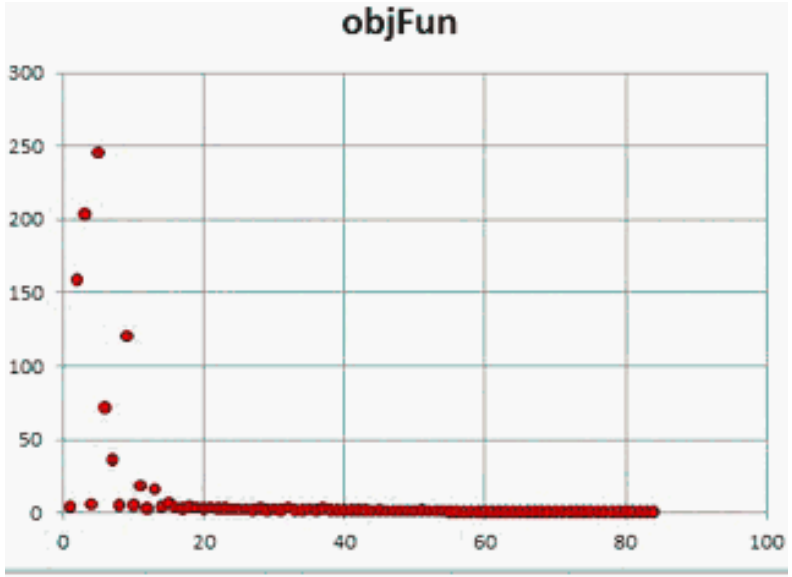


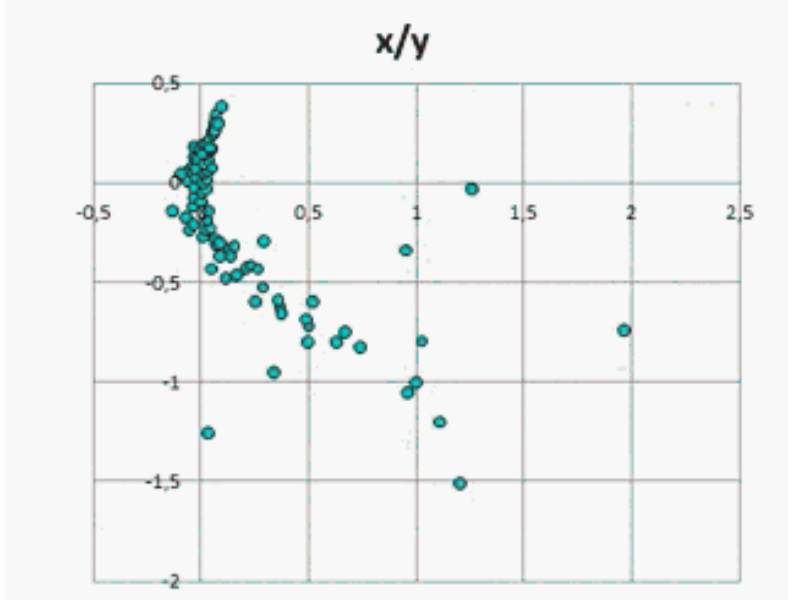


Nelder-Mead algoritmus

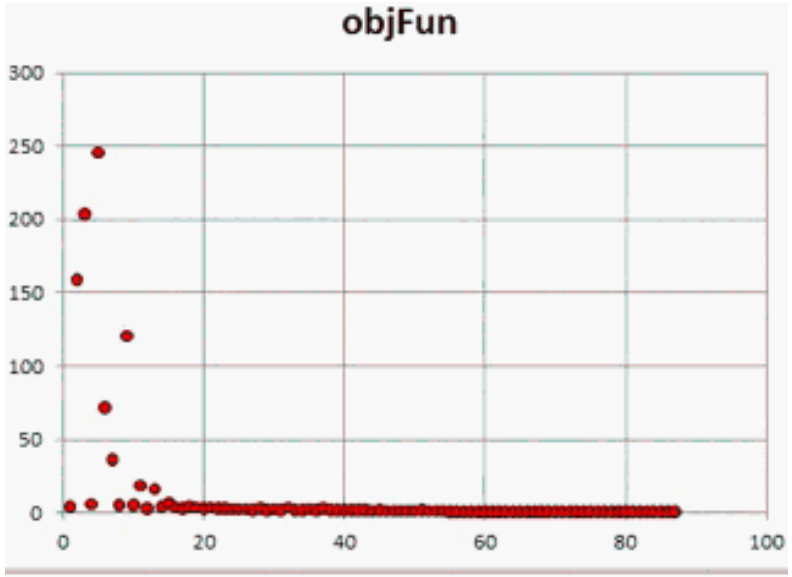


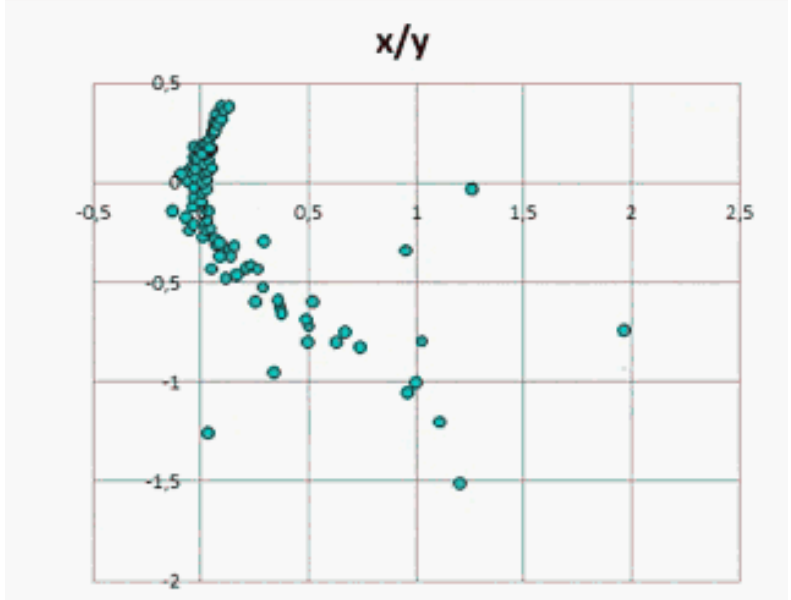
Nelder-Mead algoritmus



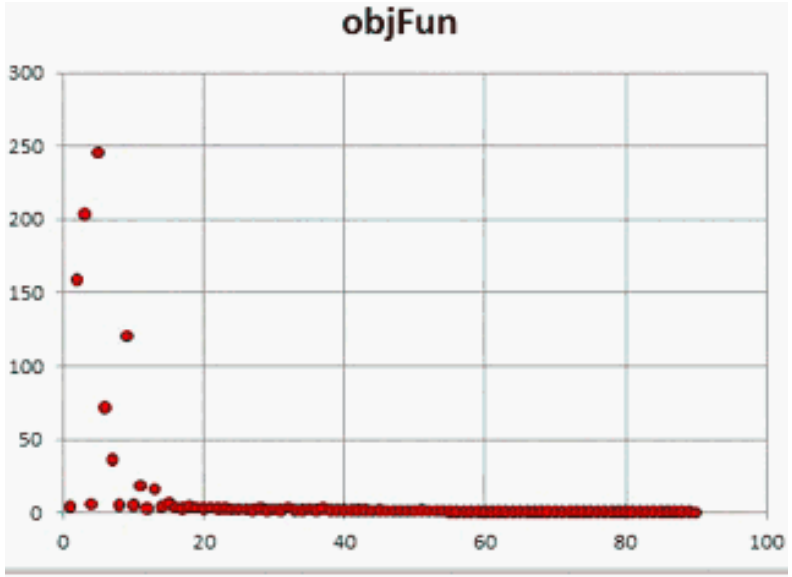


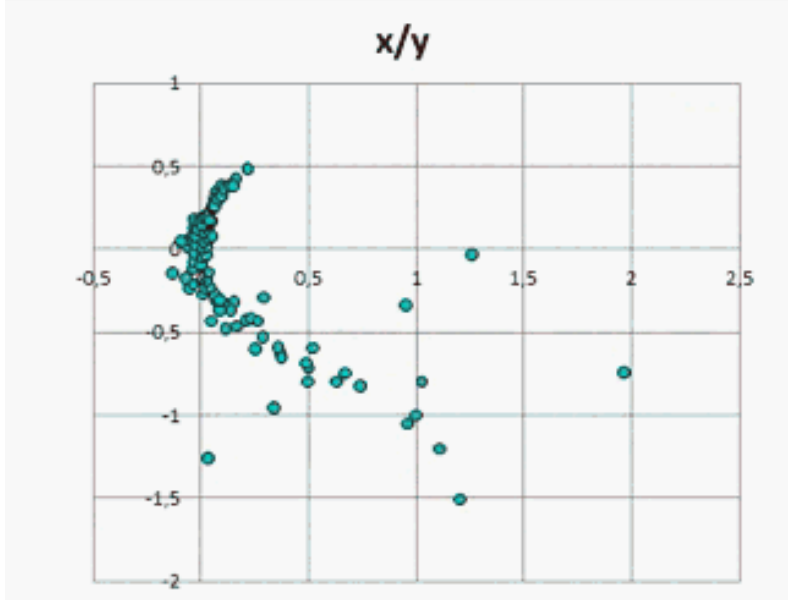
Nelder-Mead algoritmus



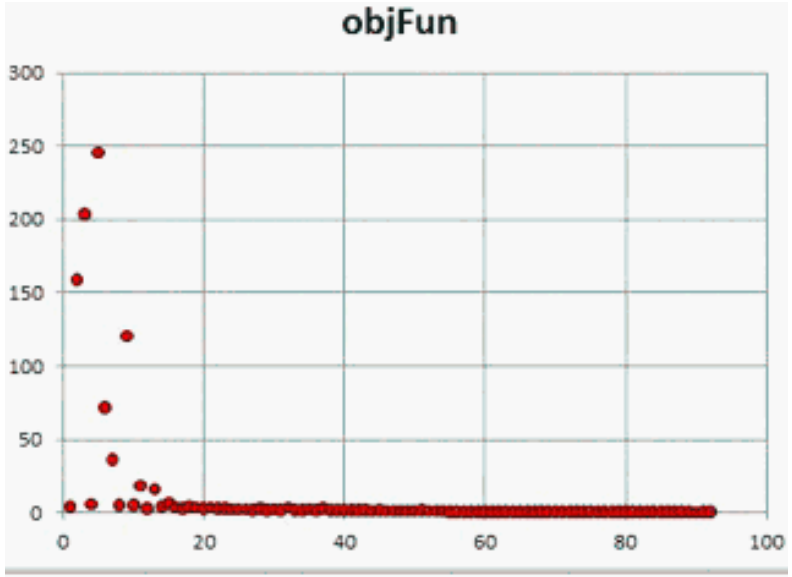


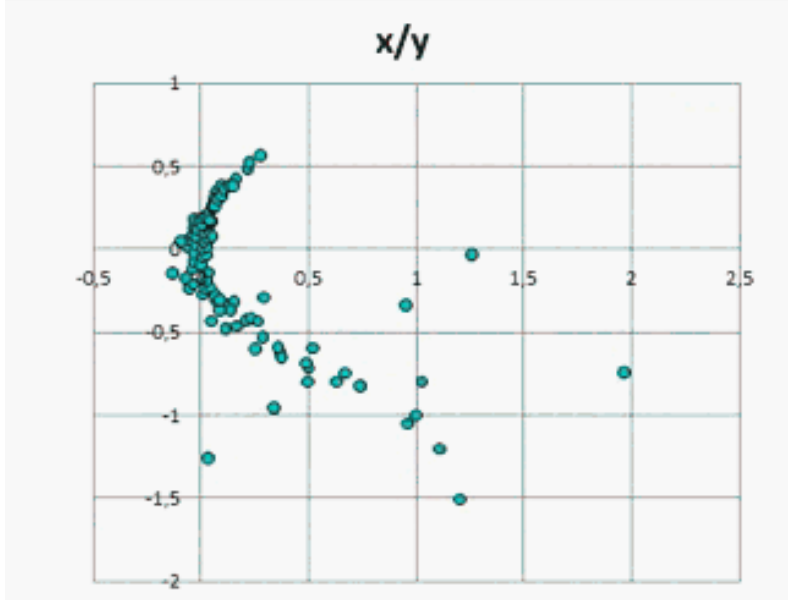
Nelder-Mead algoritmus



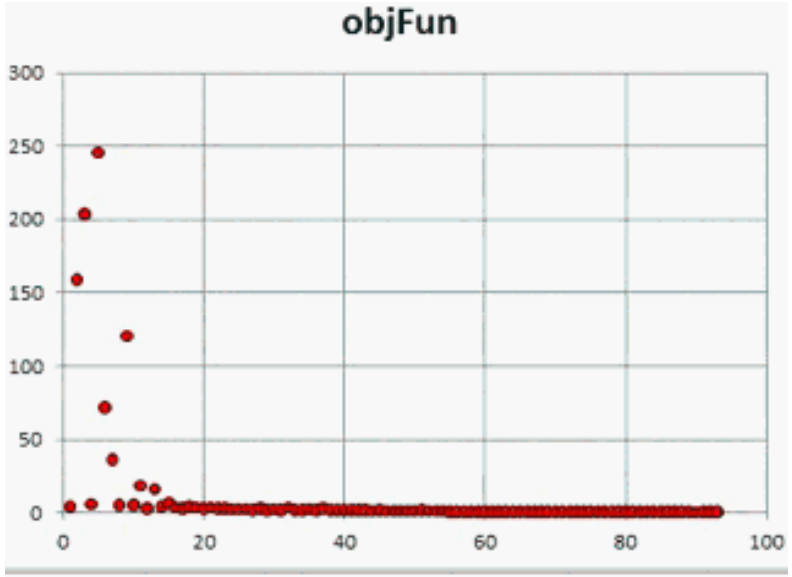


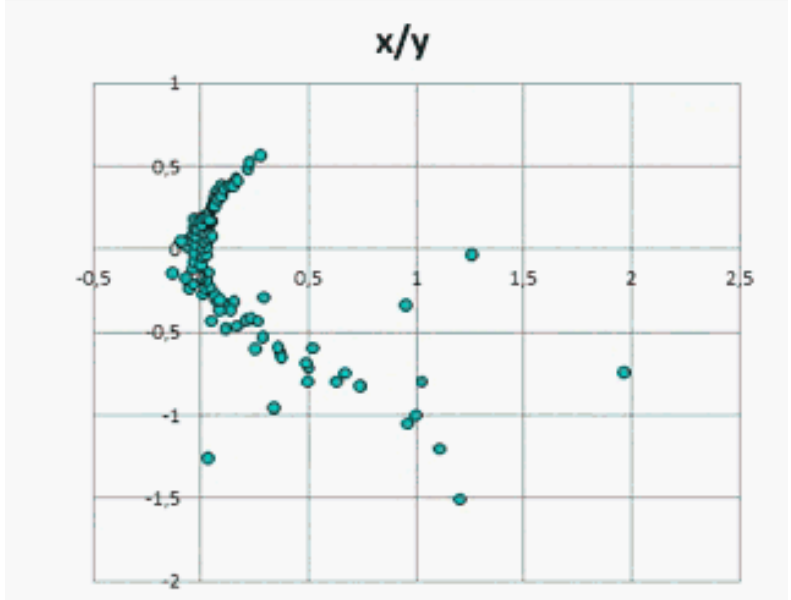
Nelder-Mead algoritmus



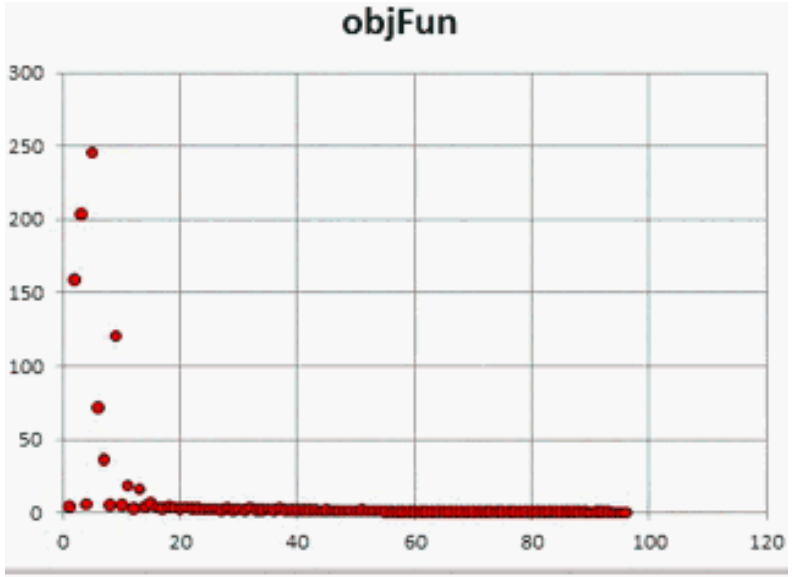


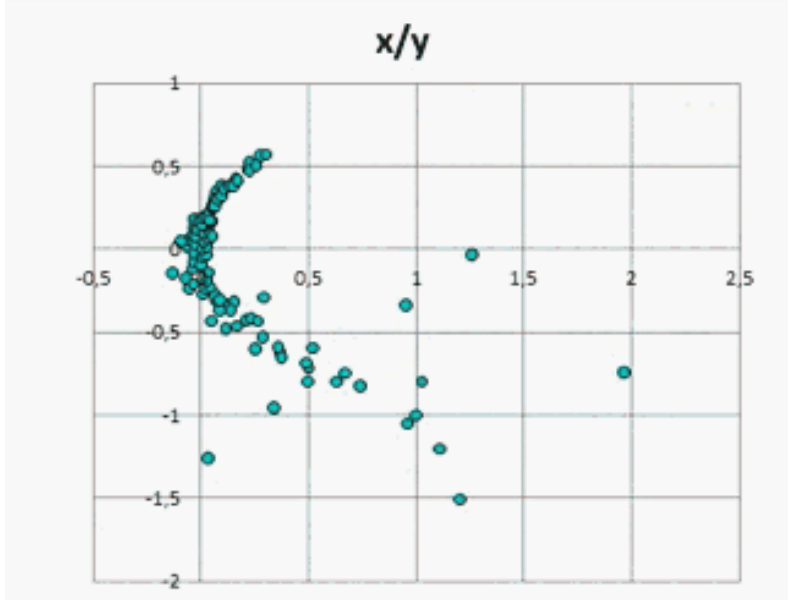
Nelder-Mead algoritmus



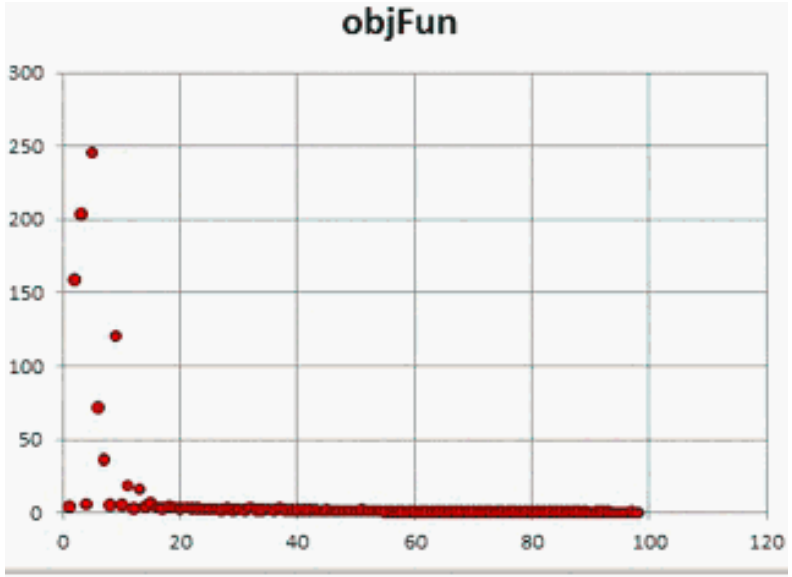


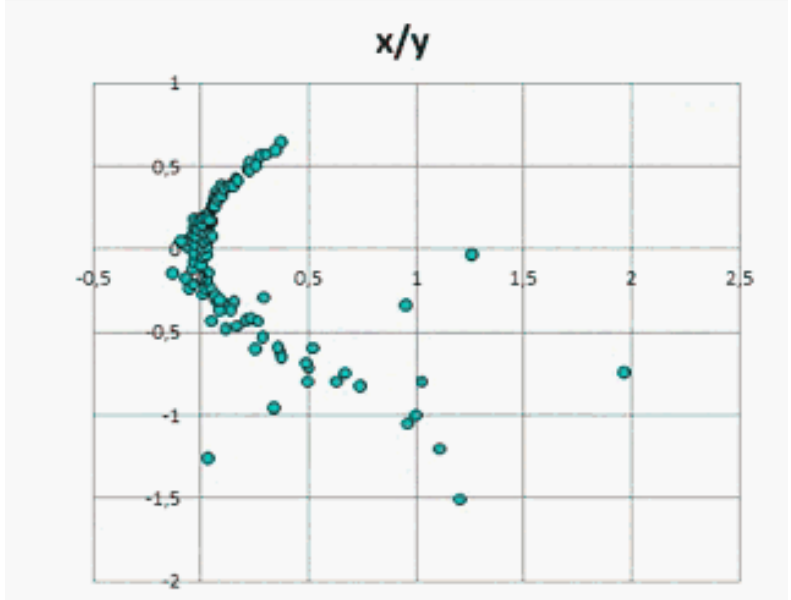
Nelder-Mead algoritmus



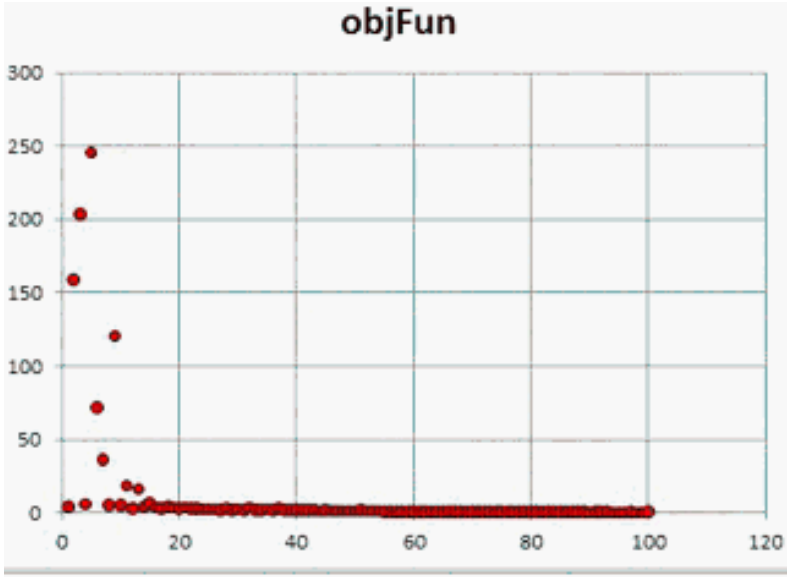


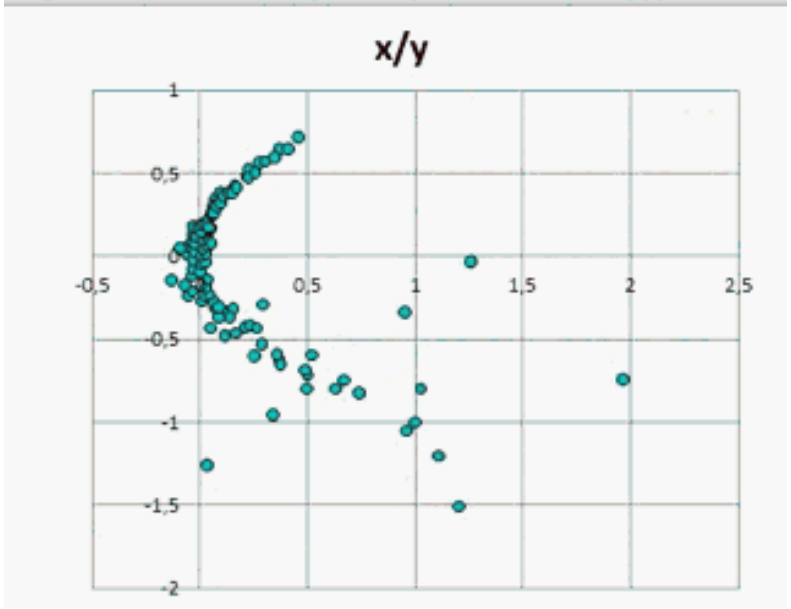
Nelder-Mead algoritmus



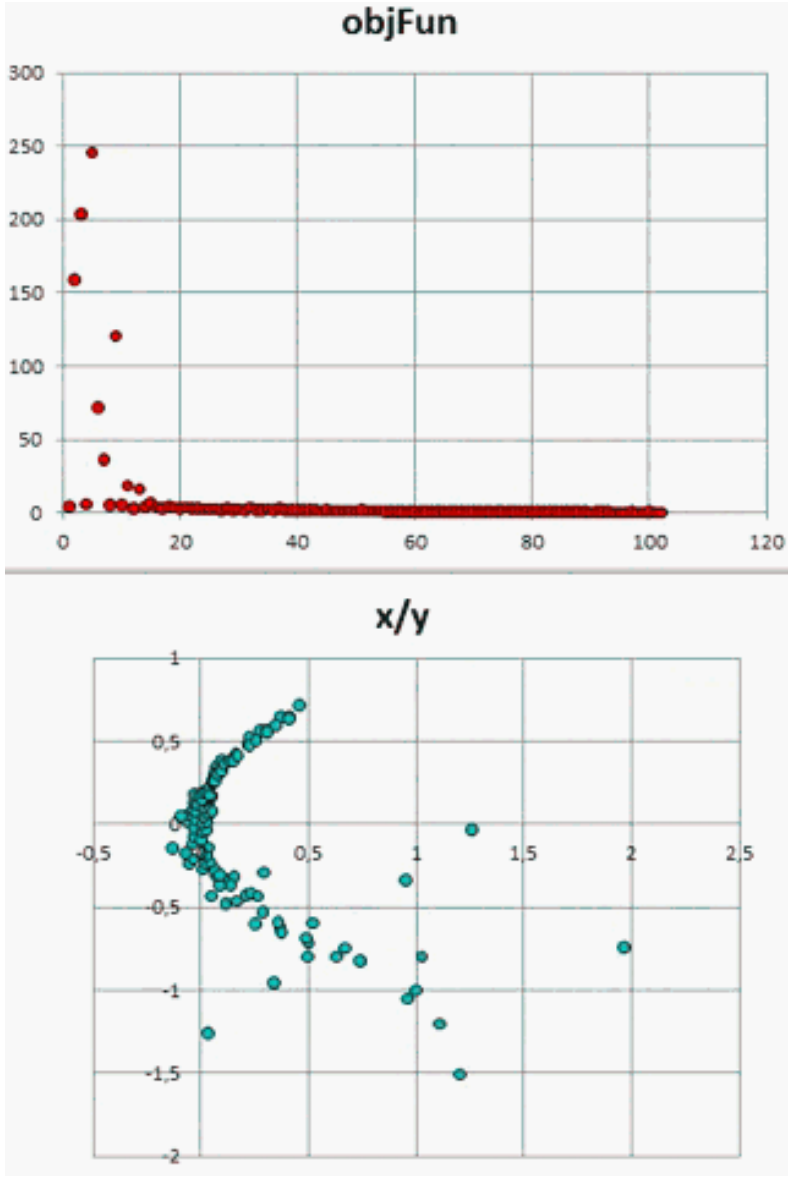


Nelder-Mead algoritmus

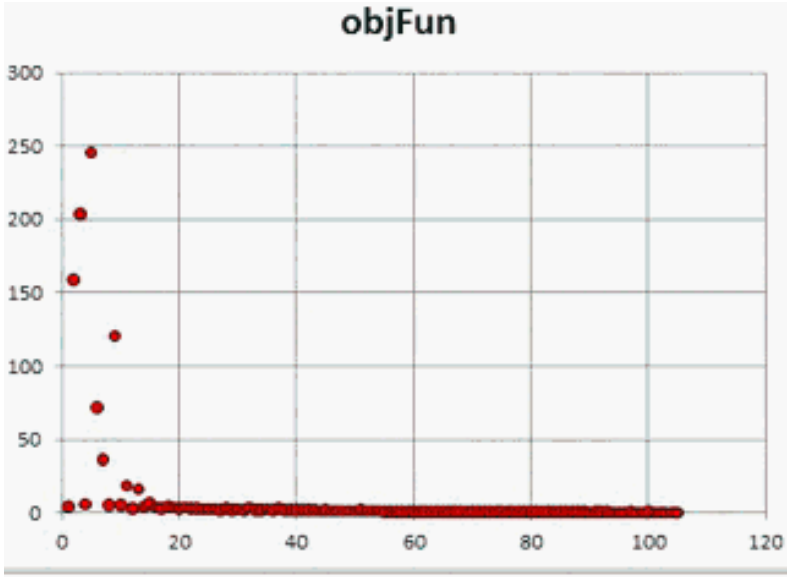


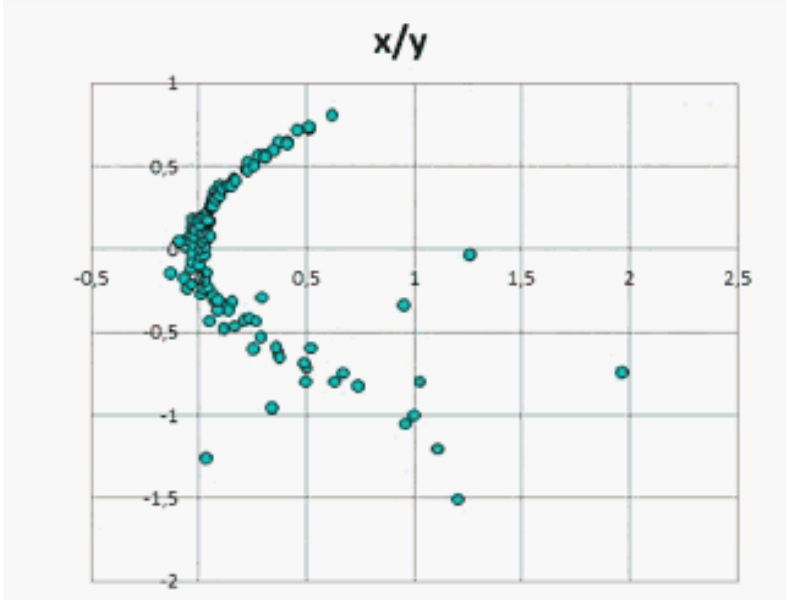


Nelder-Mead algoritmus

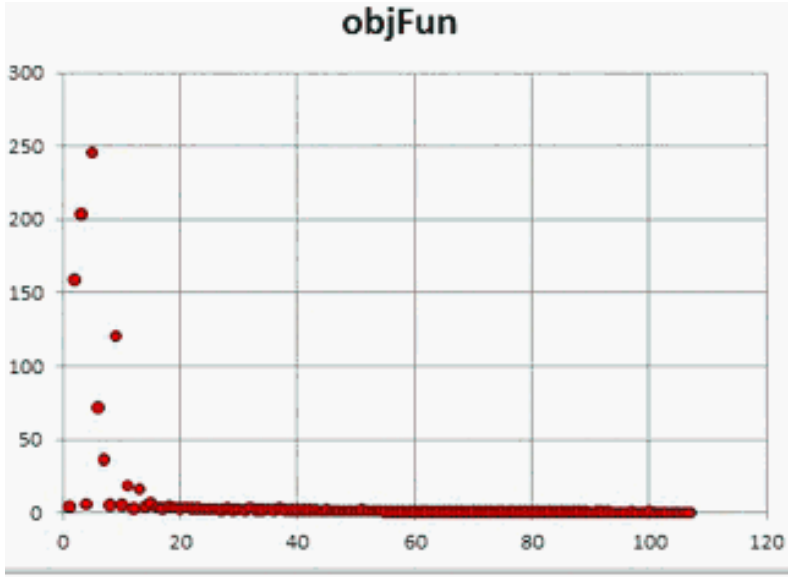


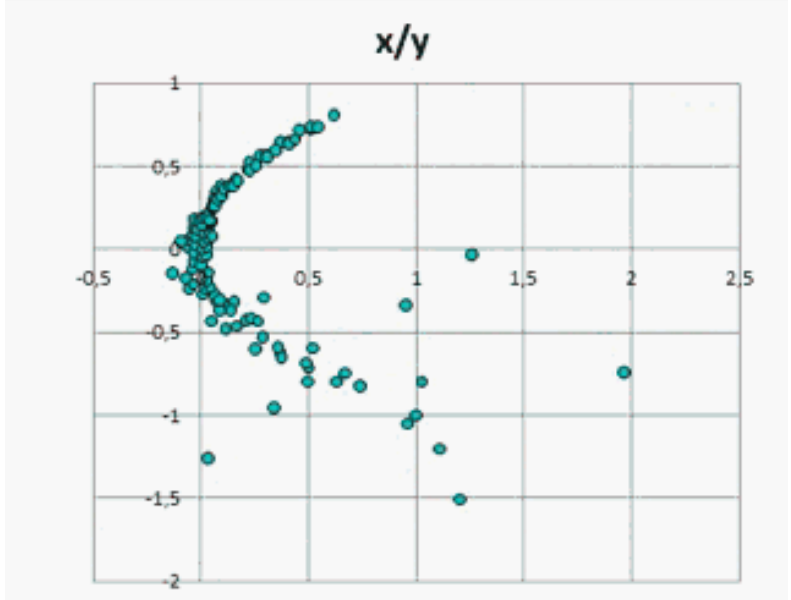
Nelder-Mead algoritmus



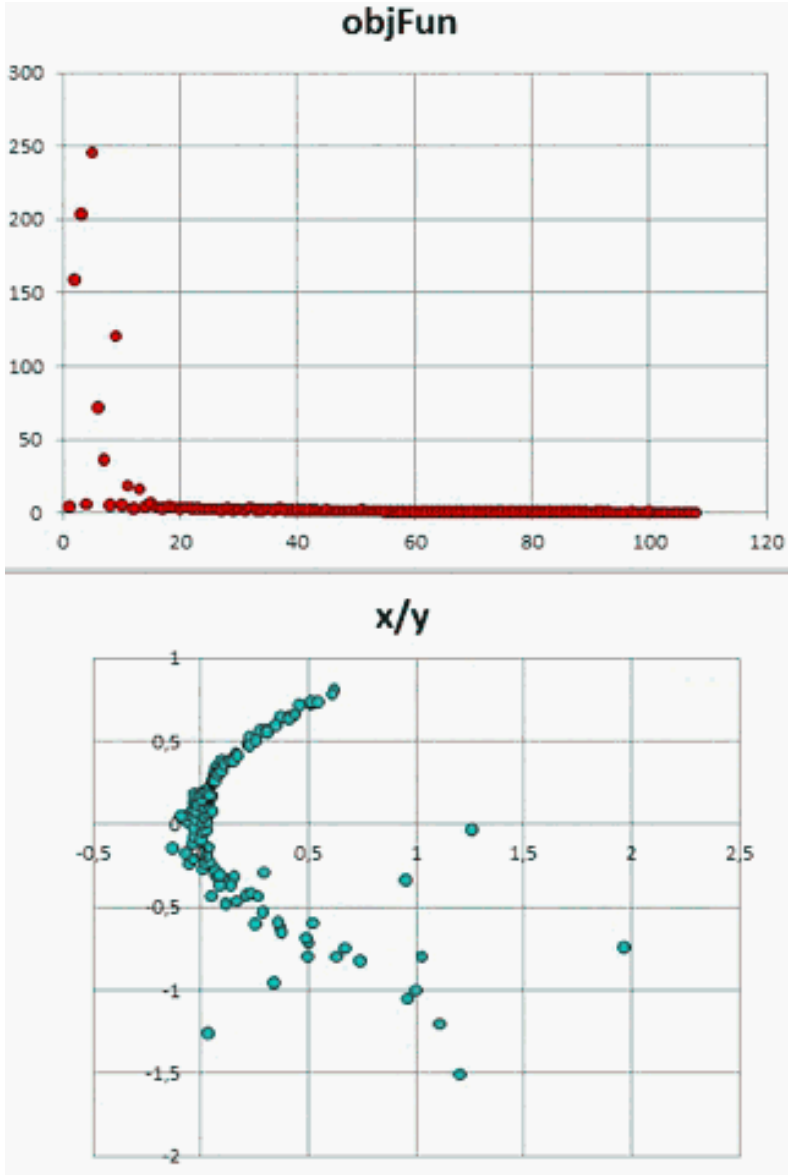


Nelder-Mead algoritmus

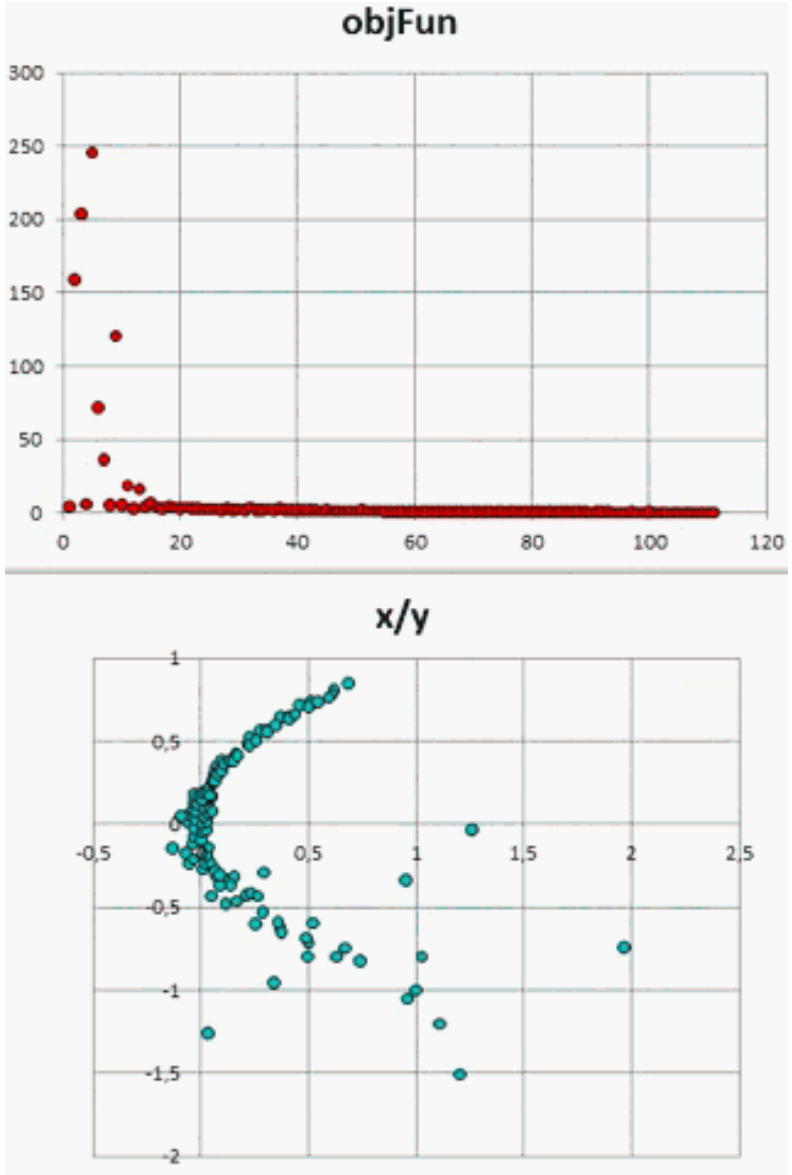




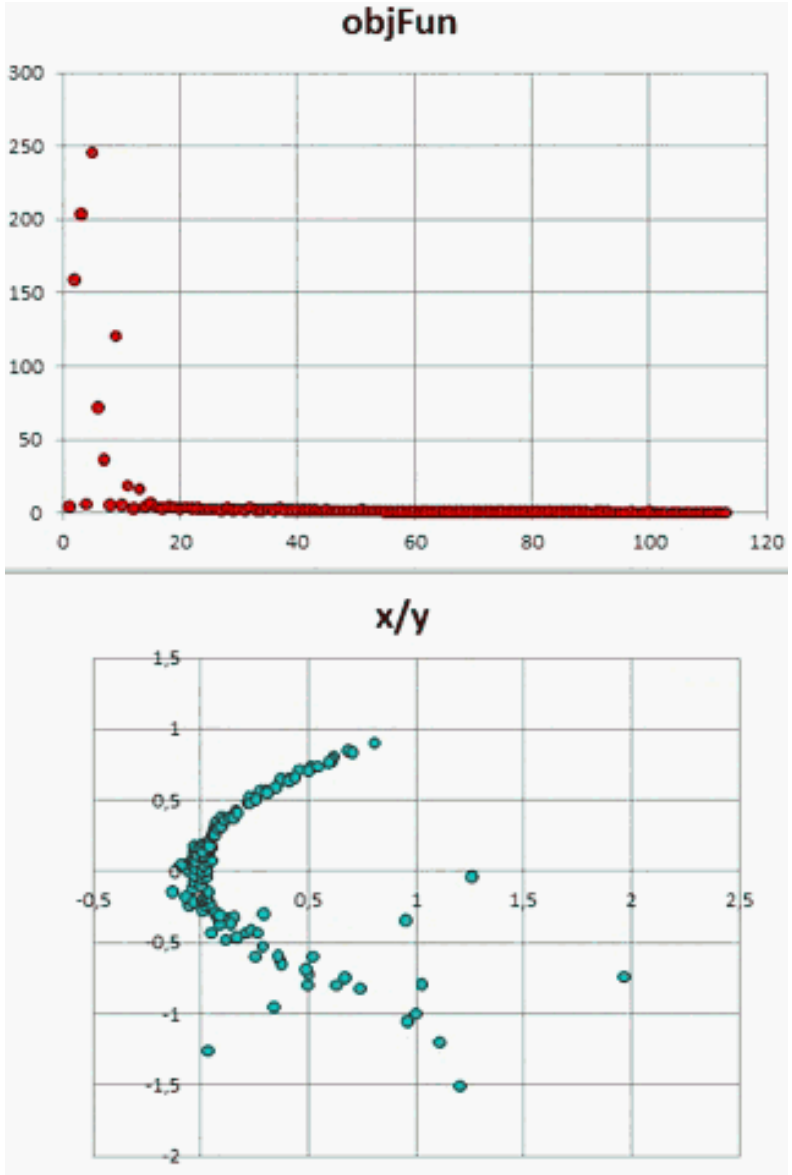
Nelder-Mead algoritmus



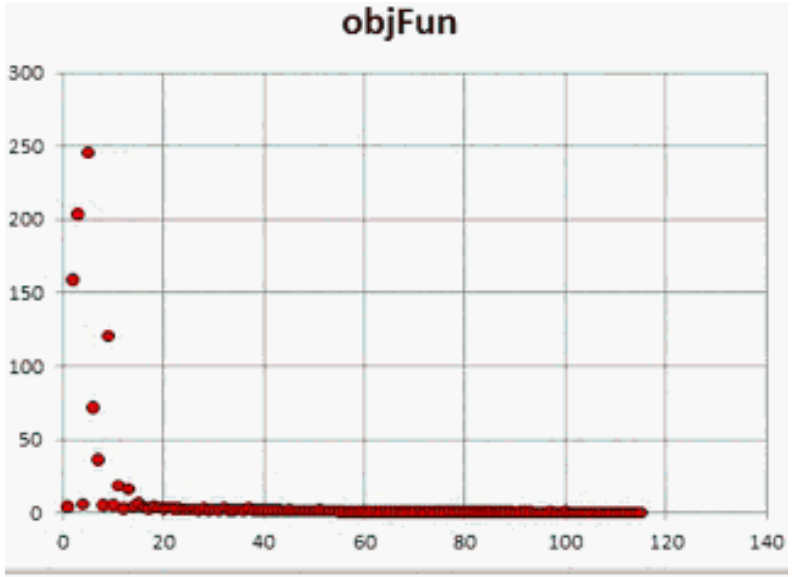
Nelder-Mead algoritmus

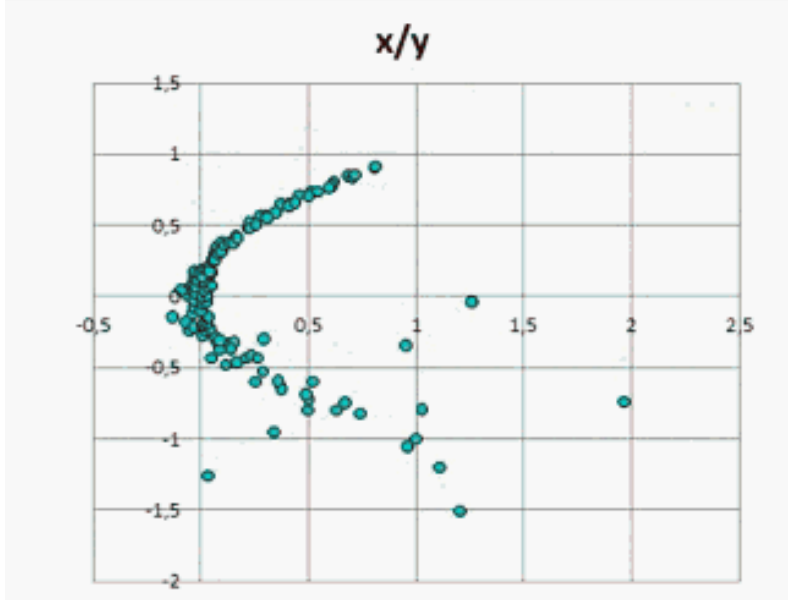


Nelder-Mead algoritmus

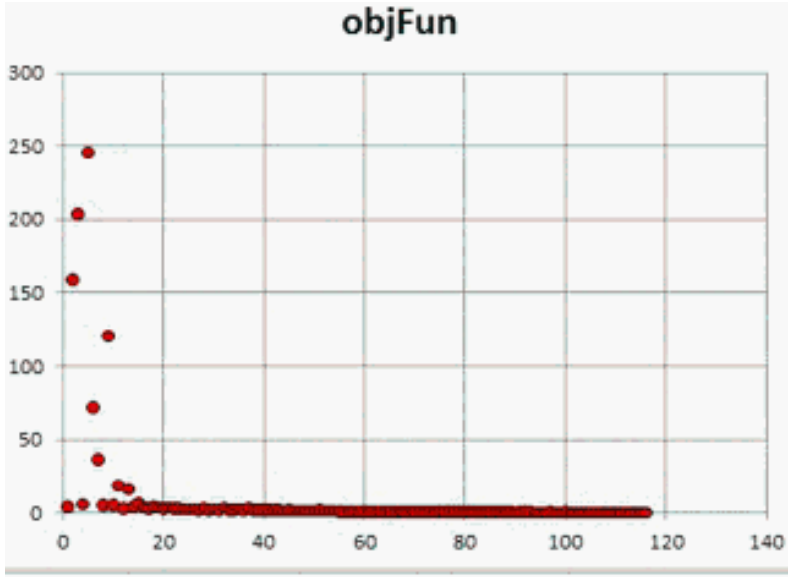


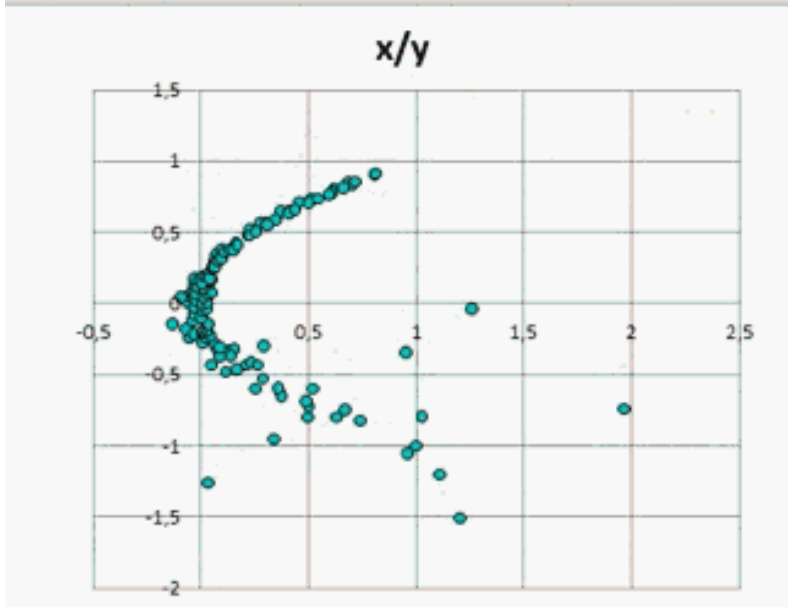
Nelder-Mead algoritmus



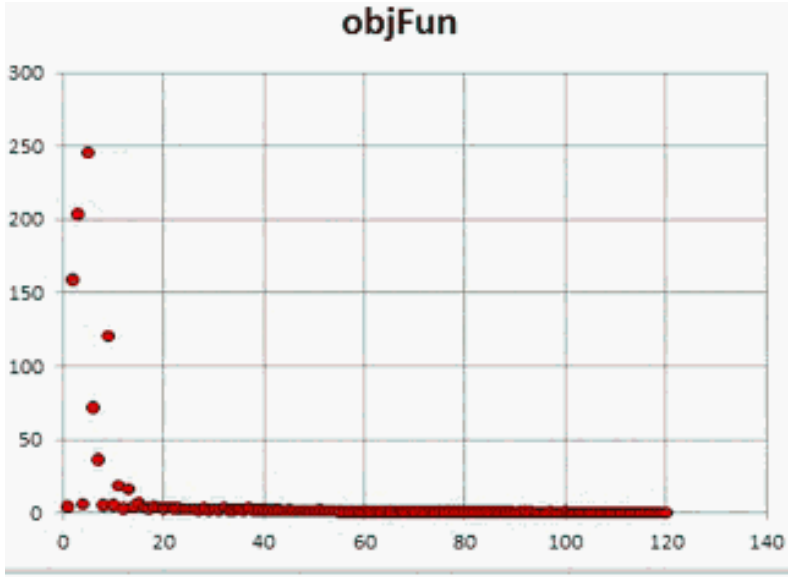


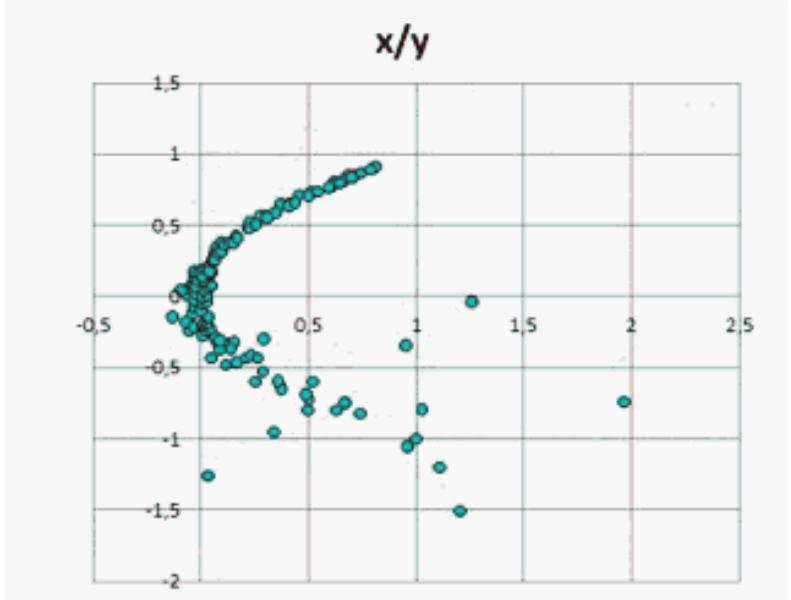
Nelder-Mead algoritmus



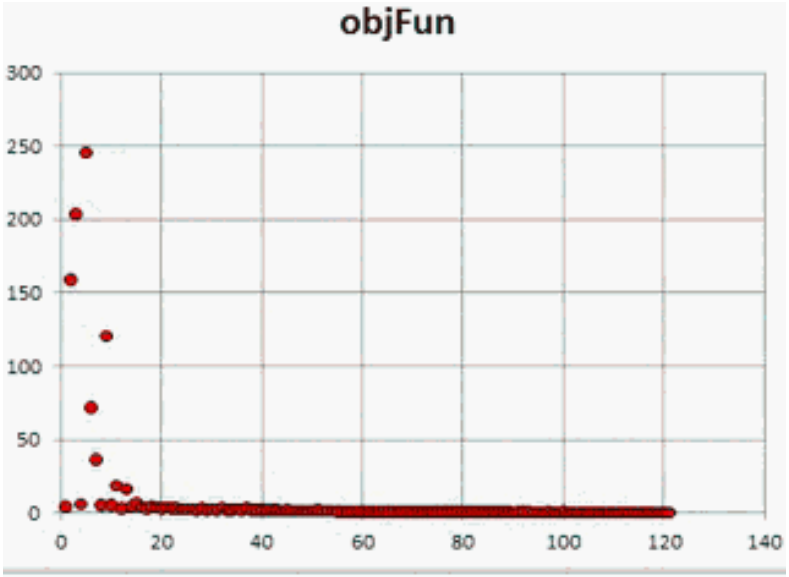


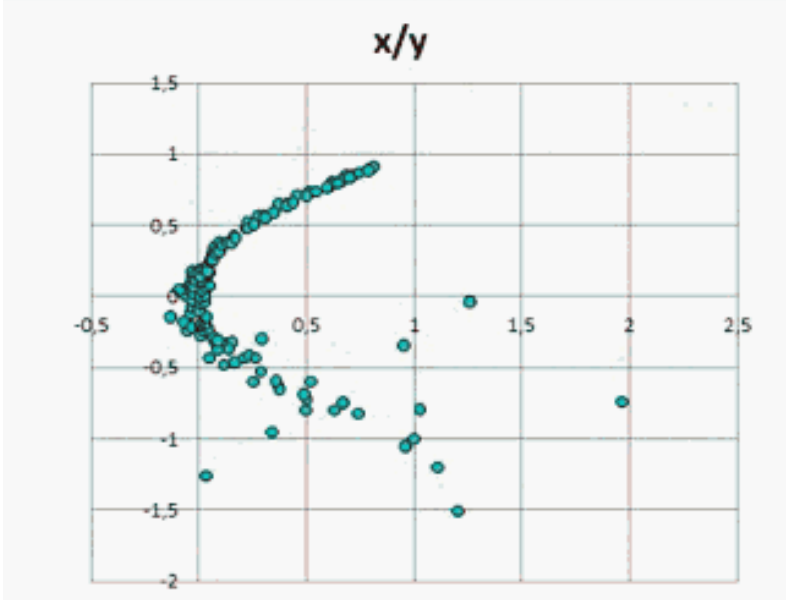
Nelder-Mead algoritmus



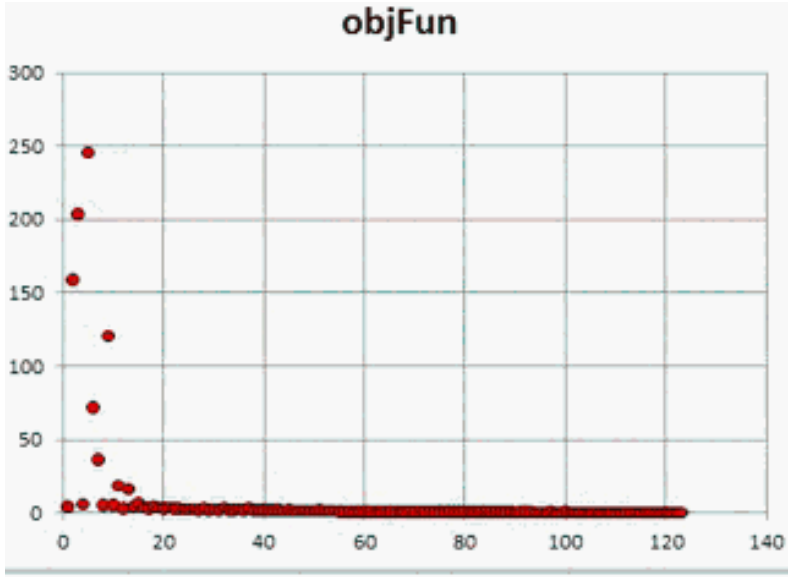


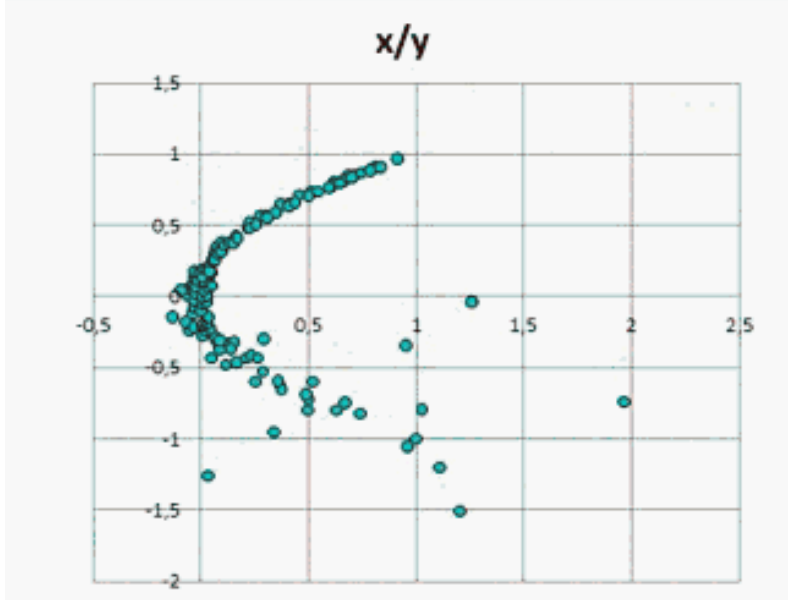
Nelder-Mead algoritmus



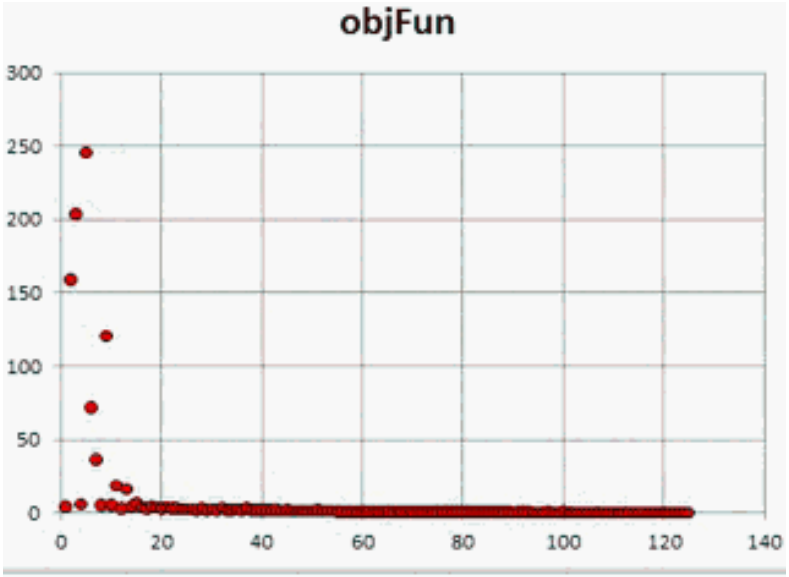


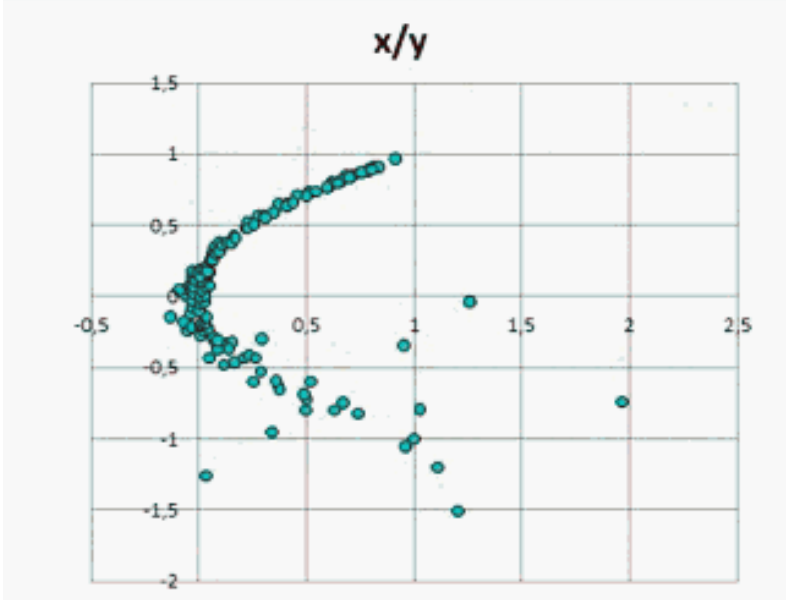
Nelder-Mead algoritmus



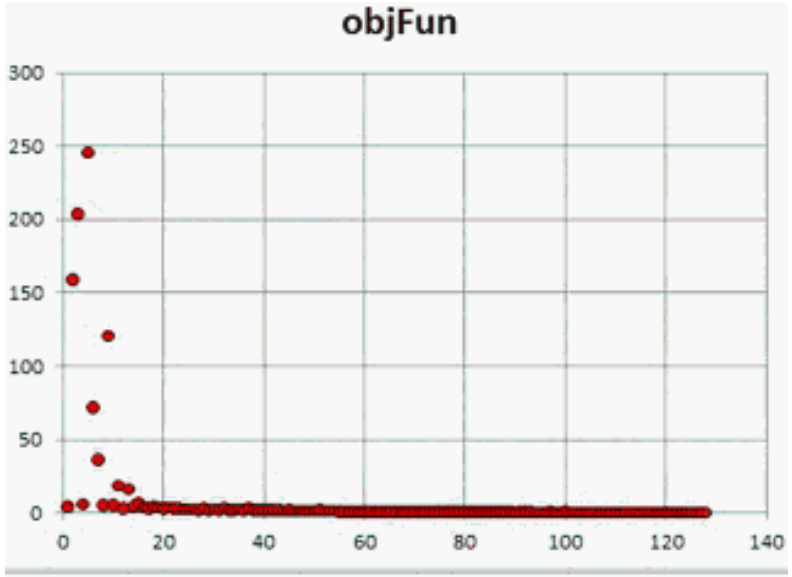


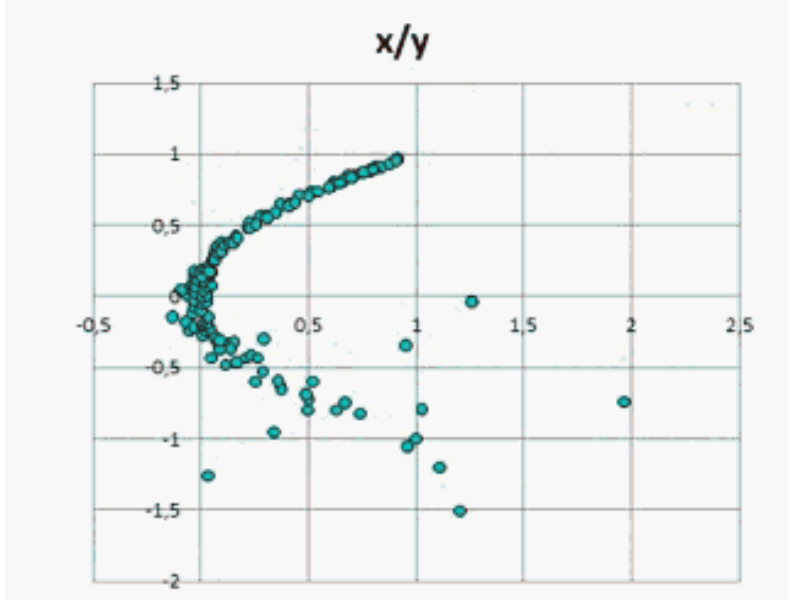
Nelder-Mead algoritmus



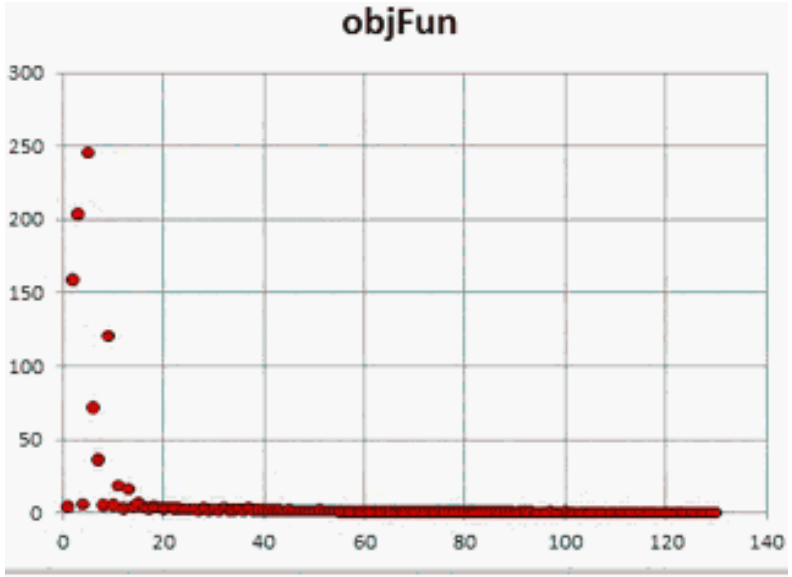


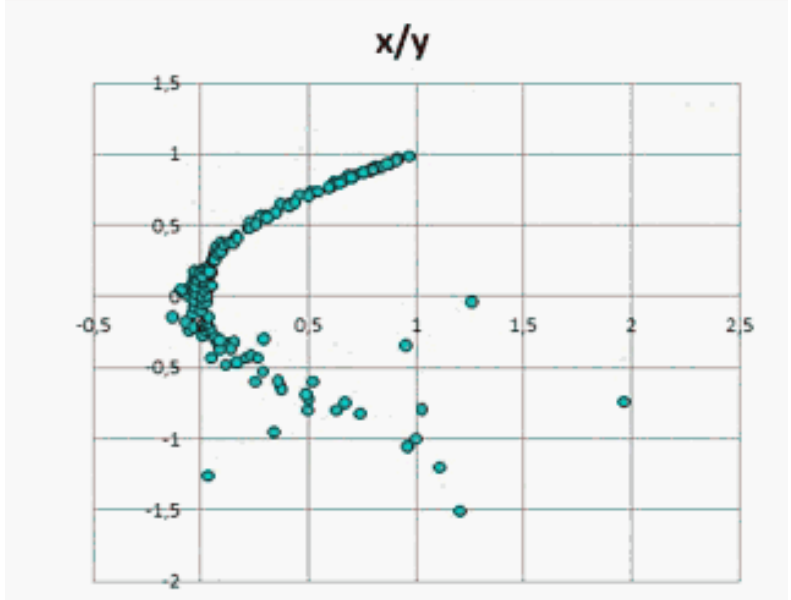
Nelder-Mead algoritmus



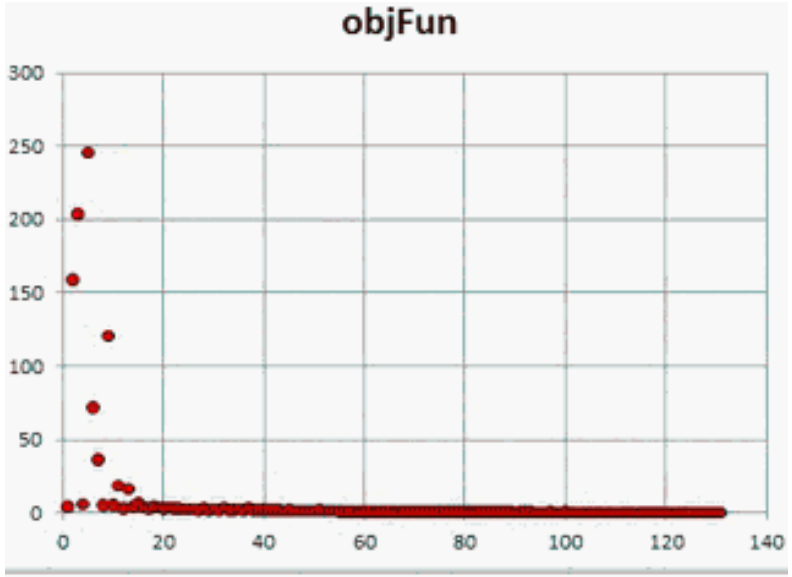


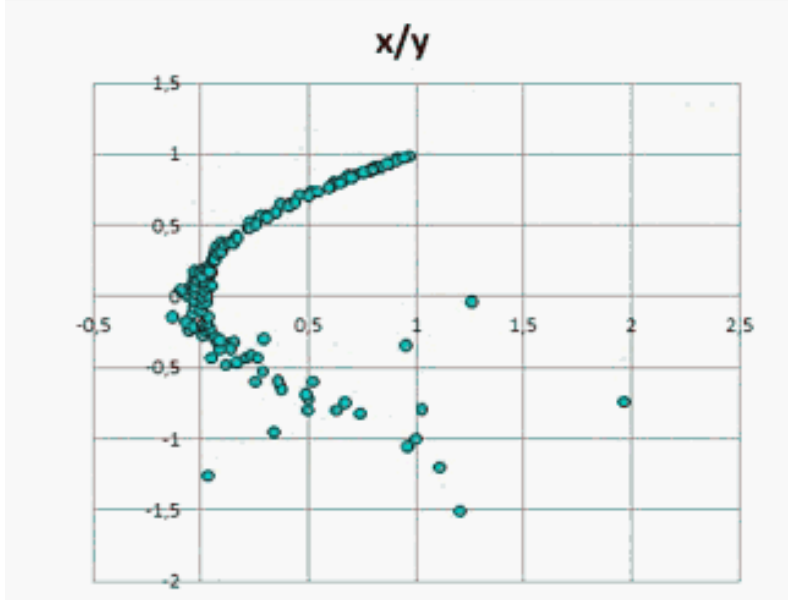
Nelder-Mead algoritmus



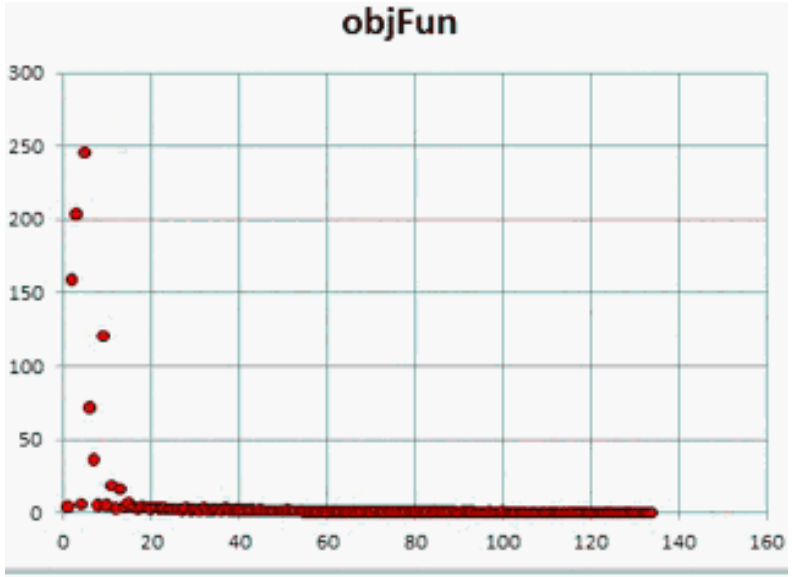


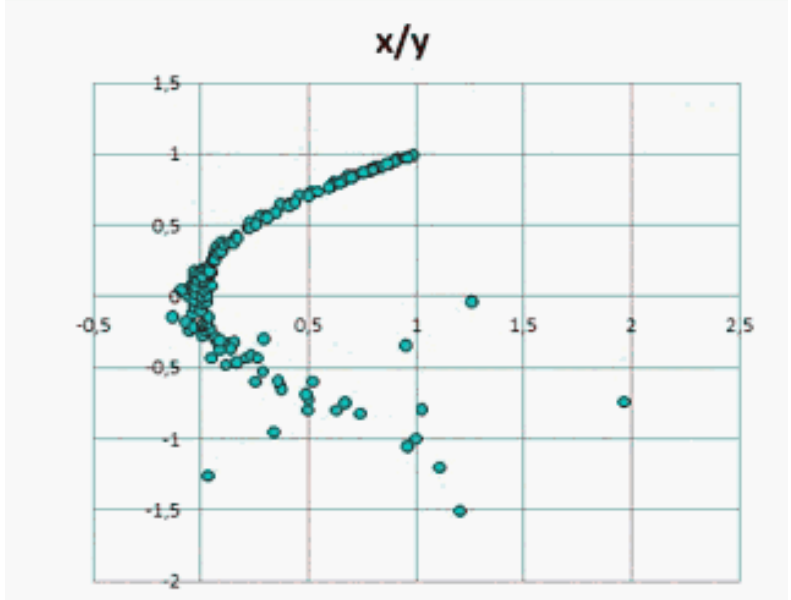
Nelder-Mead algoritmus



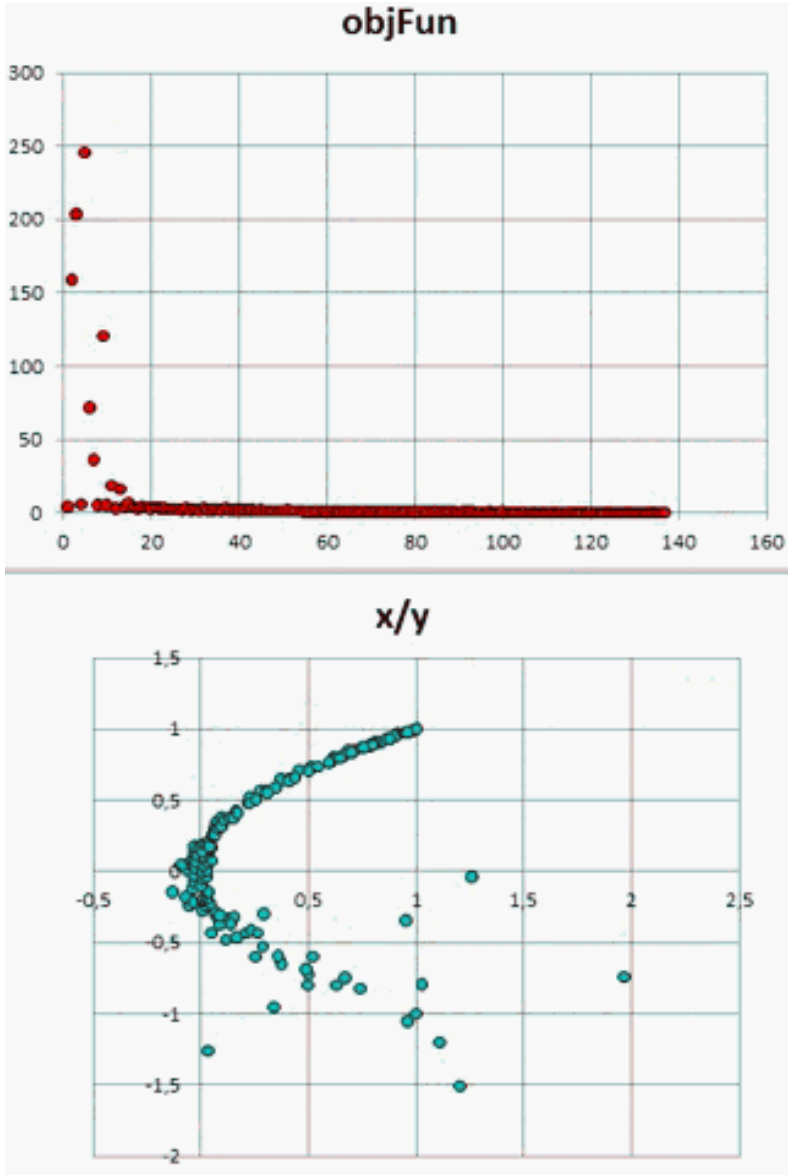


Nelder-Mead algoritmus

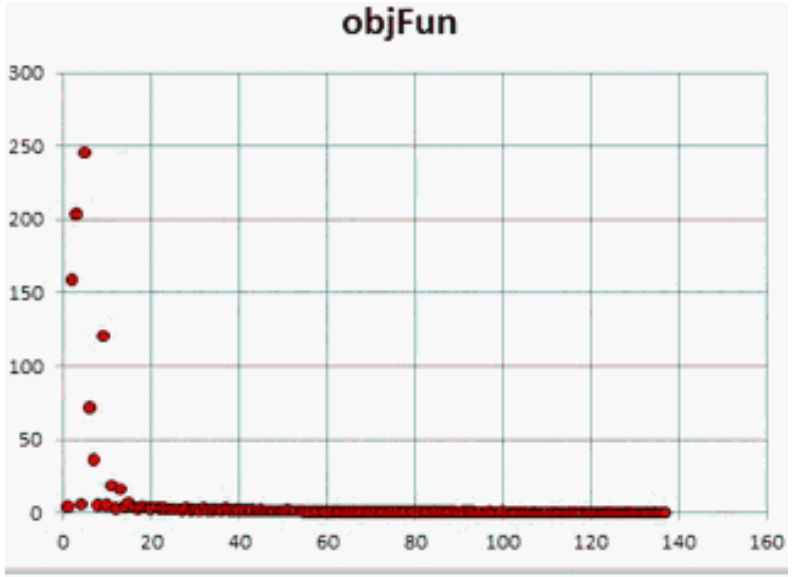


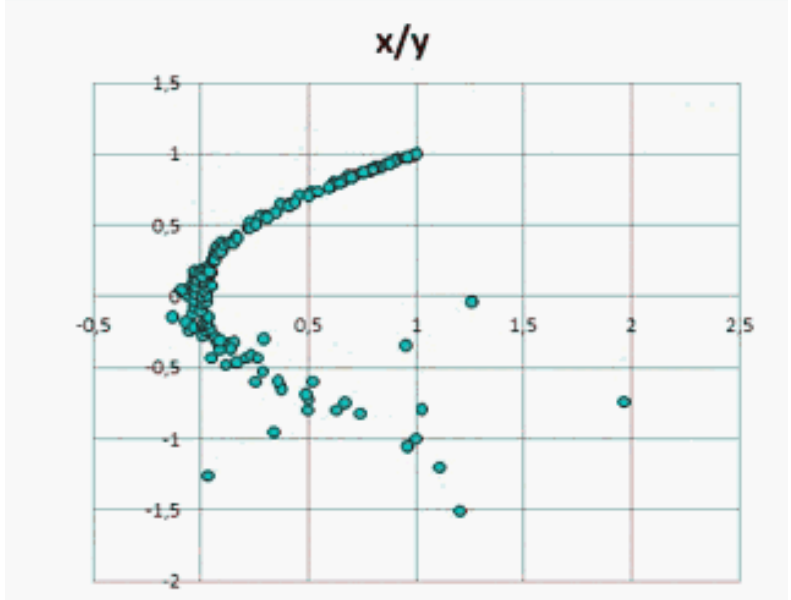


Nelder-Mead algoritmus

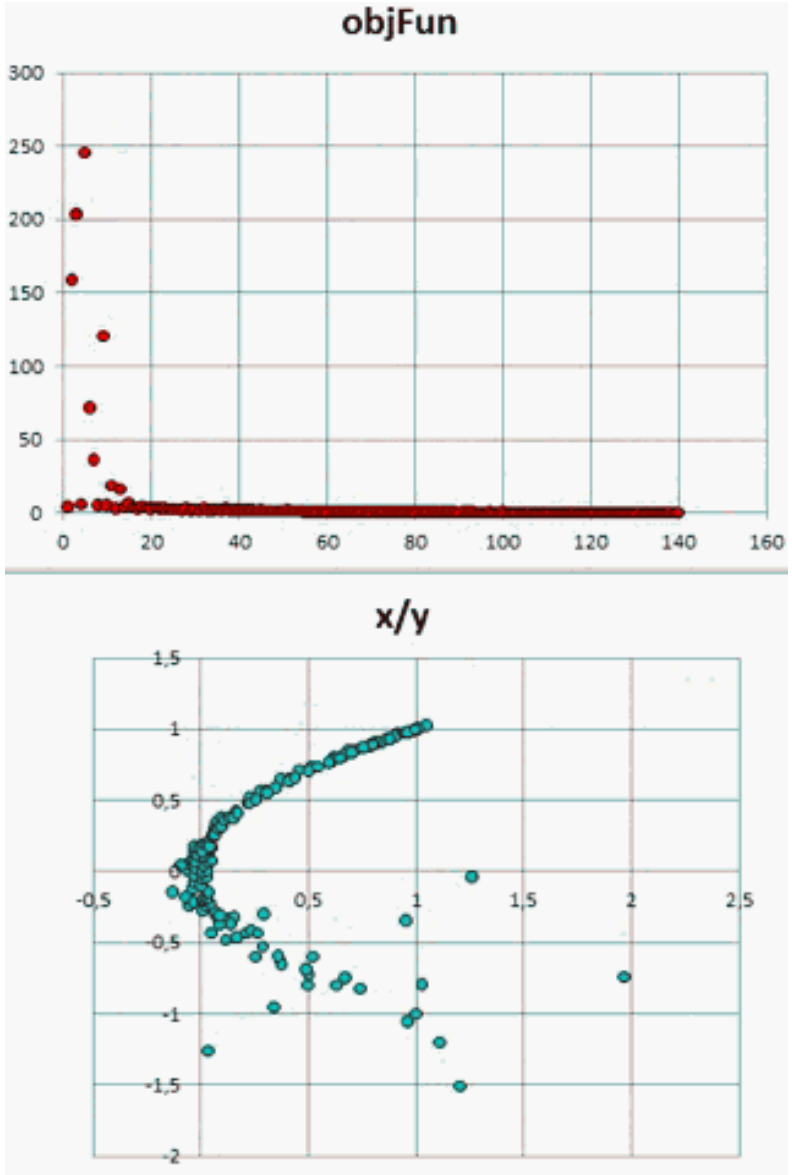


Nelder-Mead algoritmus

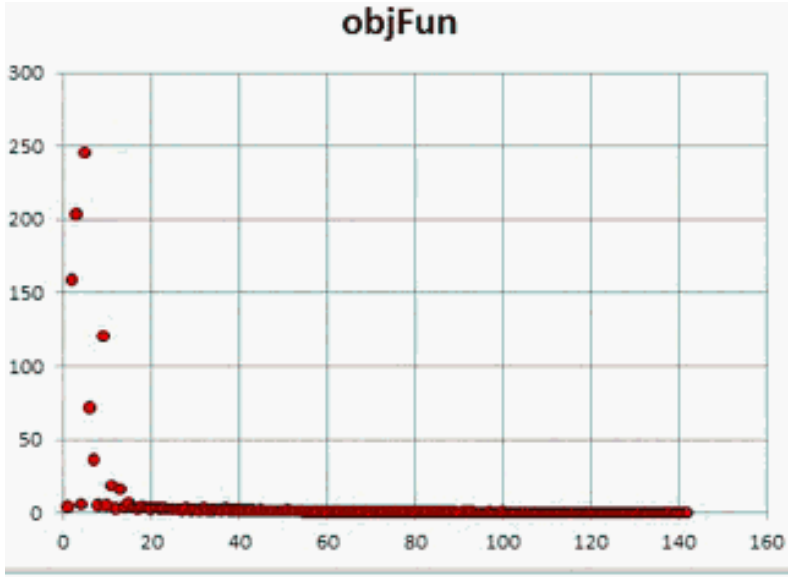


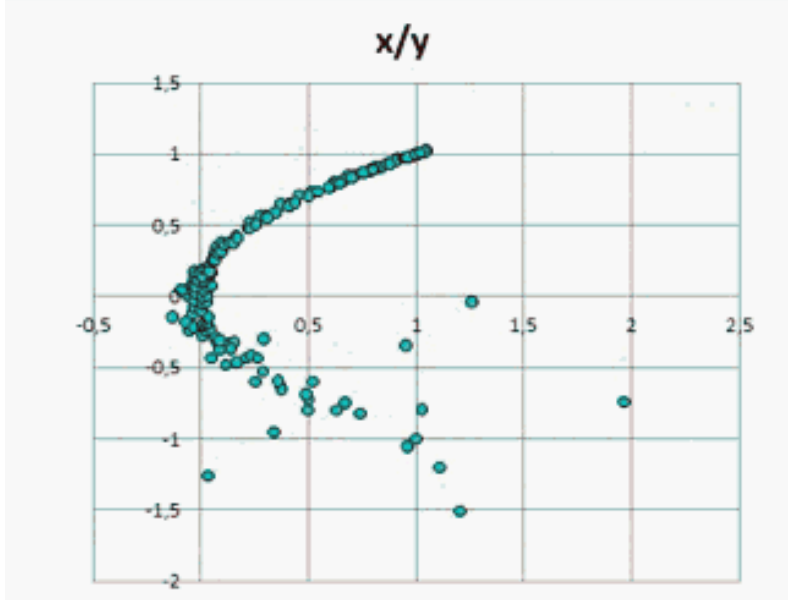


Nelder-Mead algoritmus

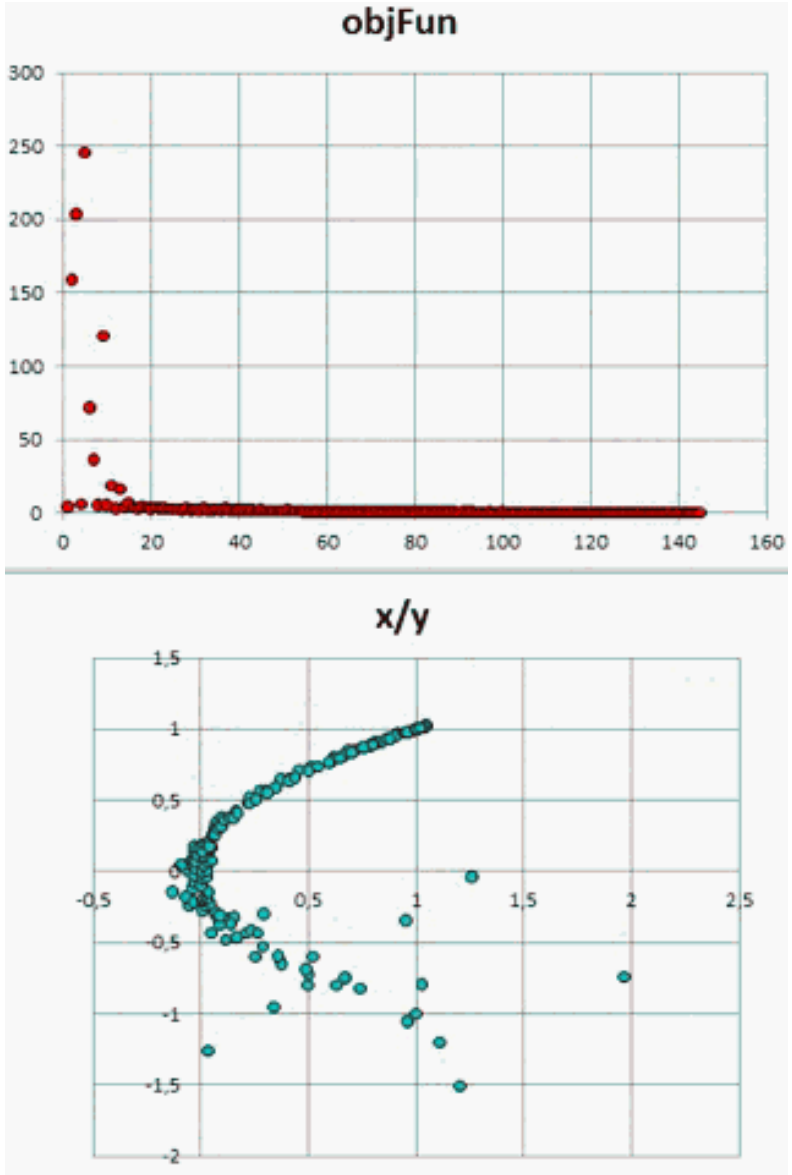


Nelder-Mead algoritmus

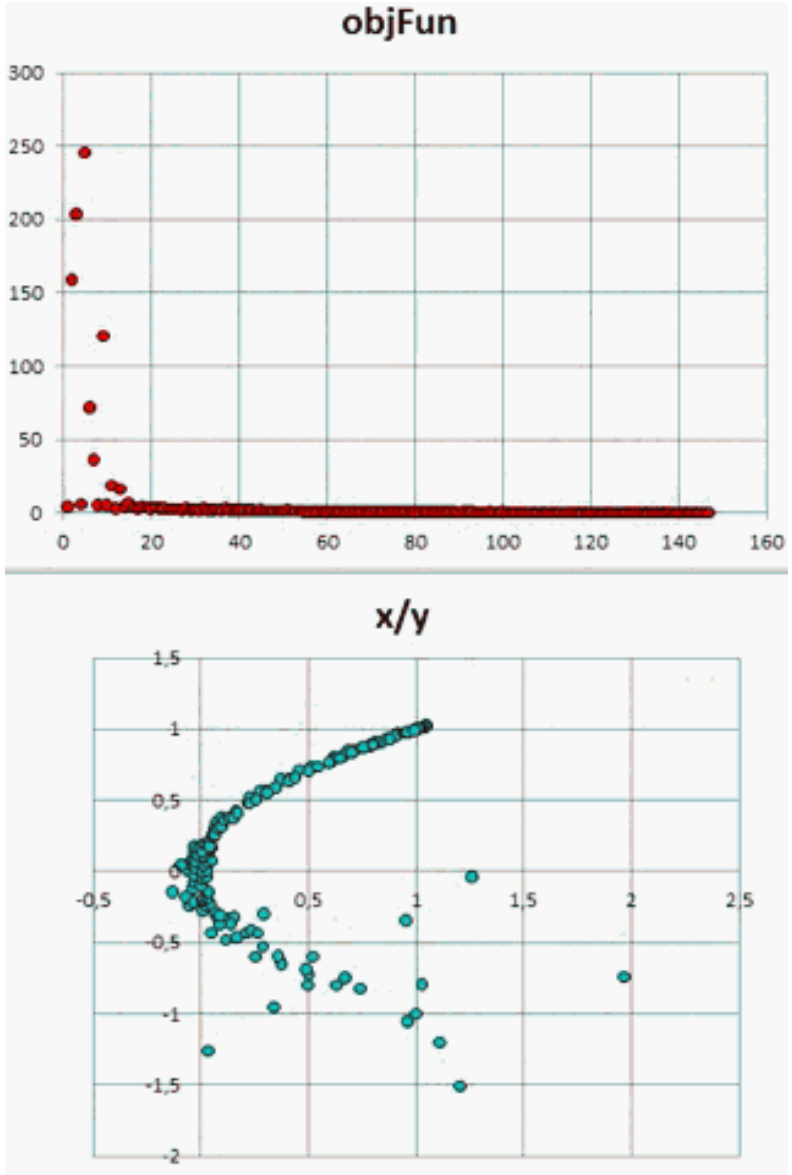




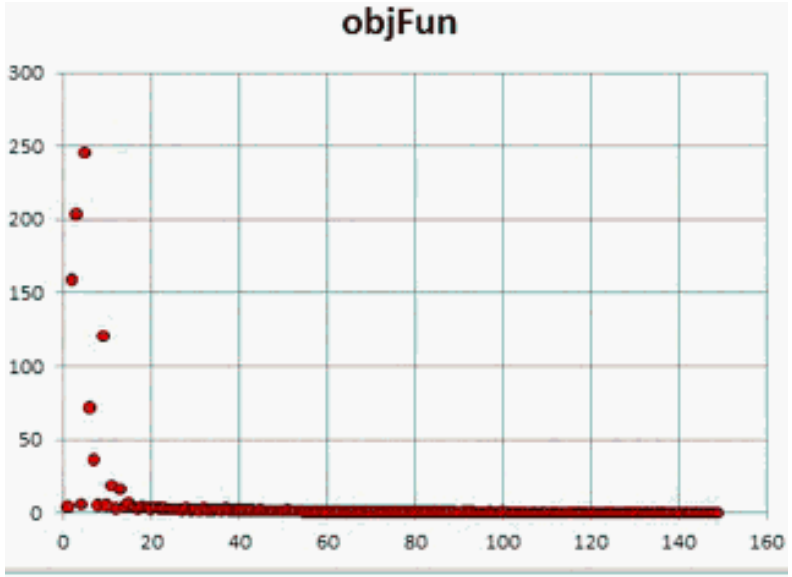
Nelder-Mead algoritmus

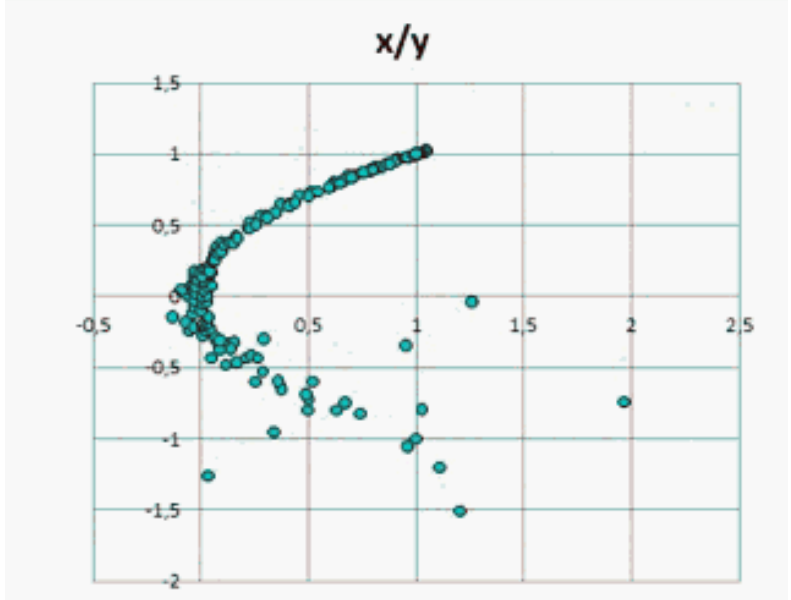


Nelder-Mead algoritmus

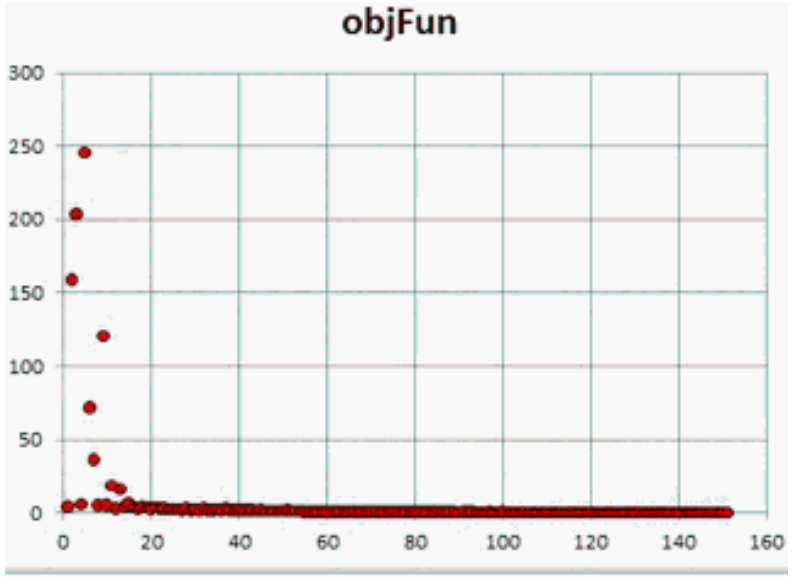


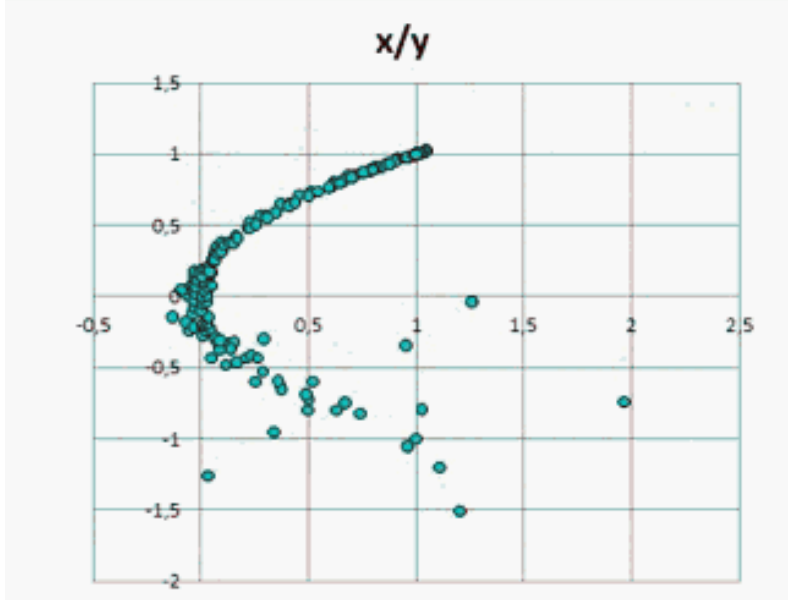
Nelder-Mead algoritmus



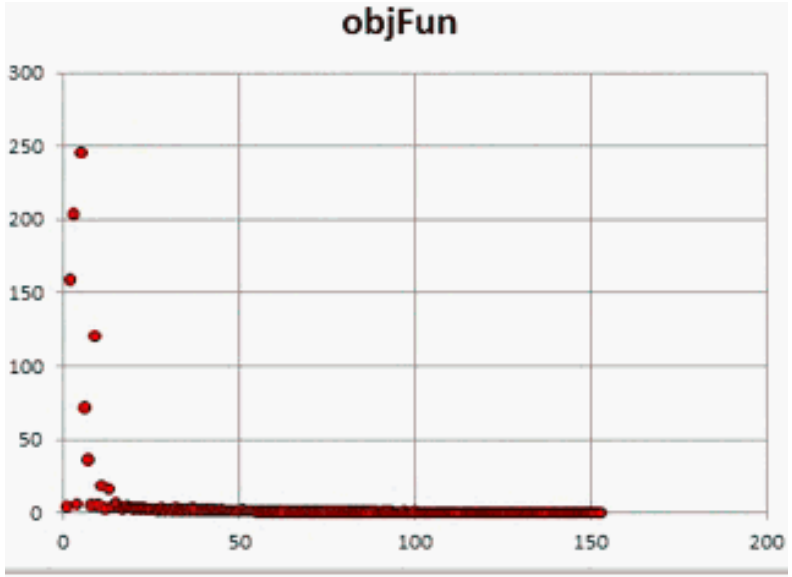


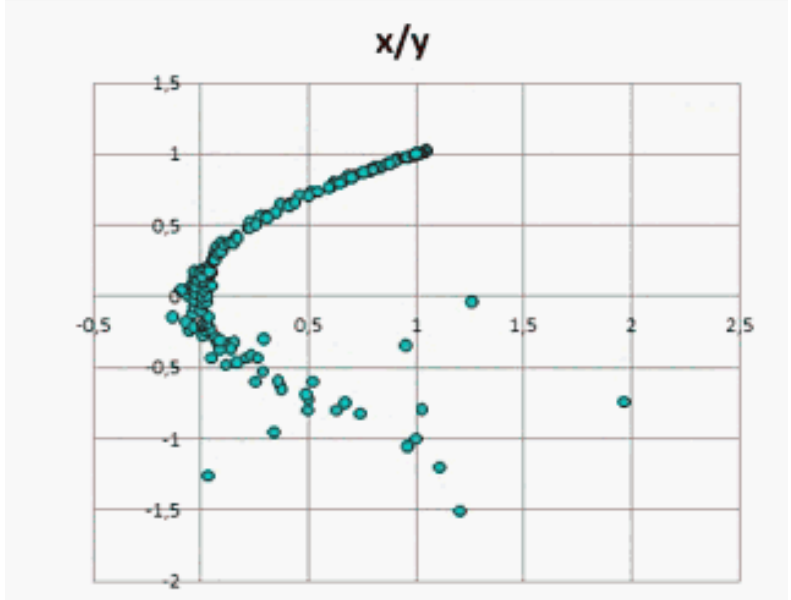
Nelder-Mead algoritmus



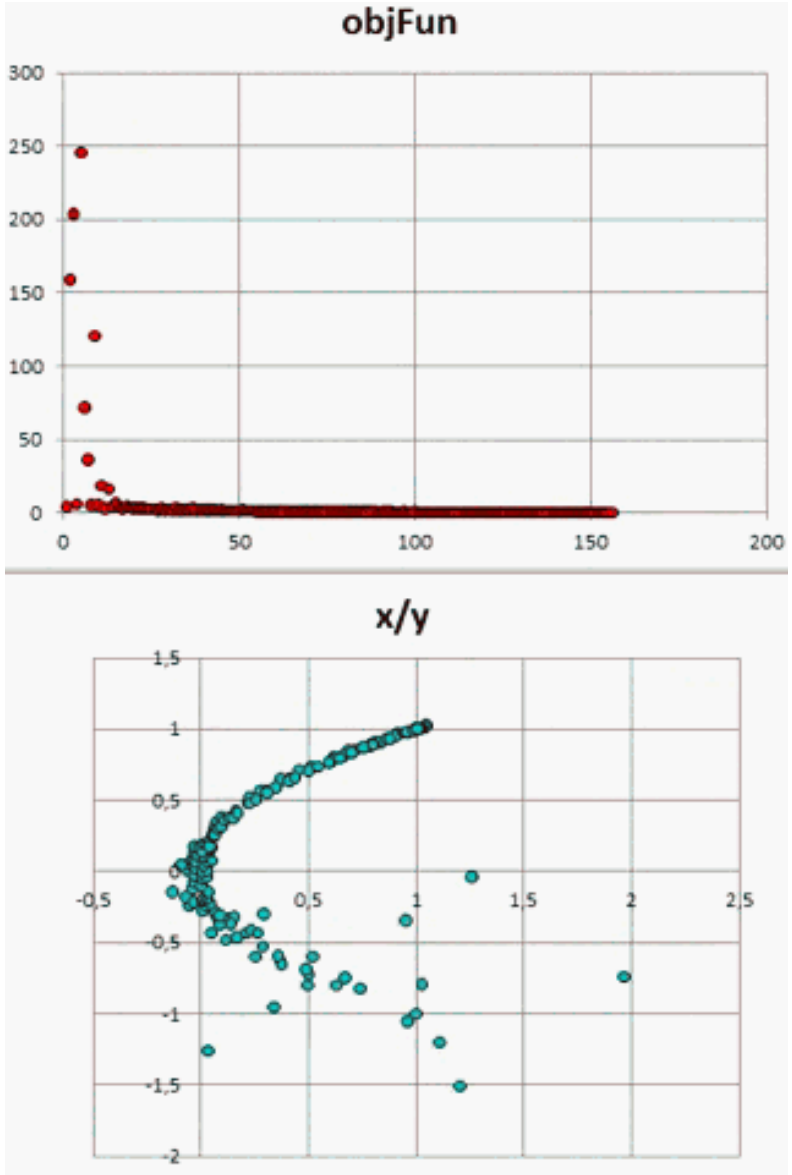


Nelder-Mead algoritmus

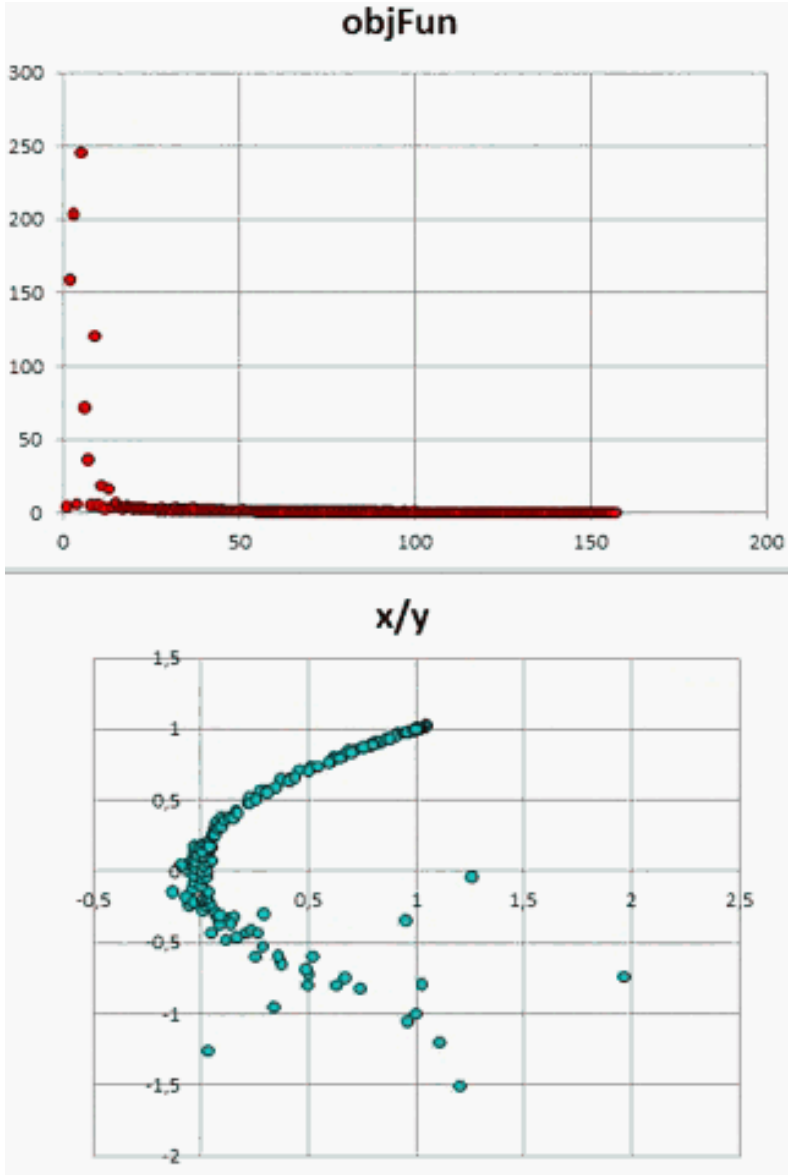




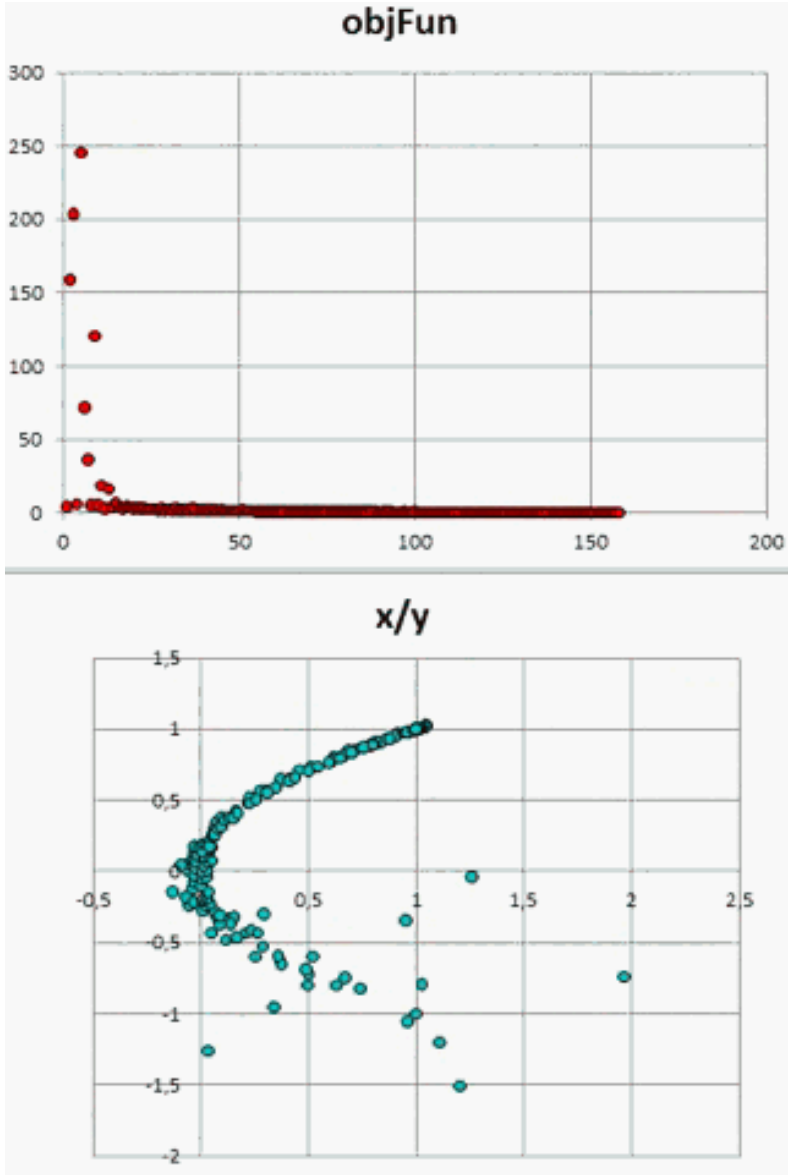
Nelder-Mead algoritmus



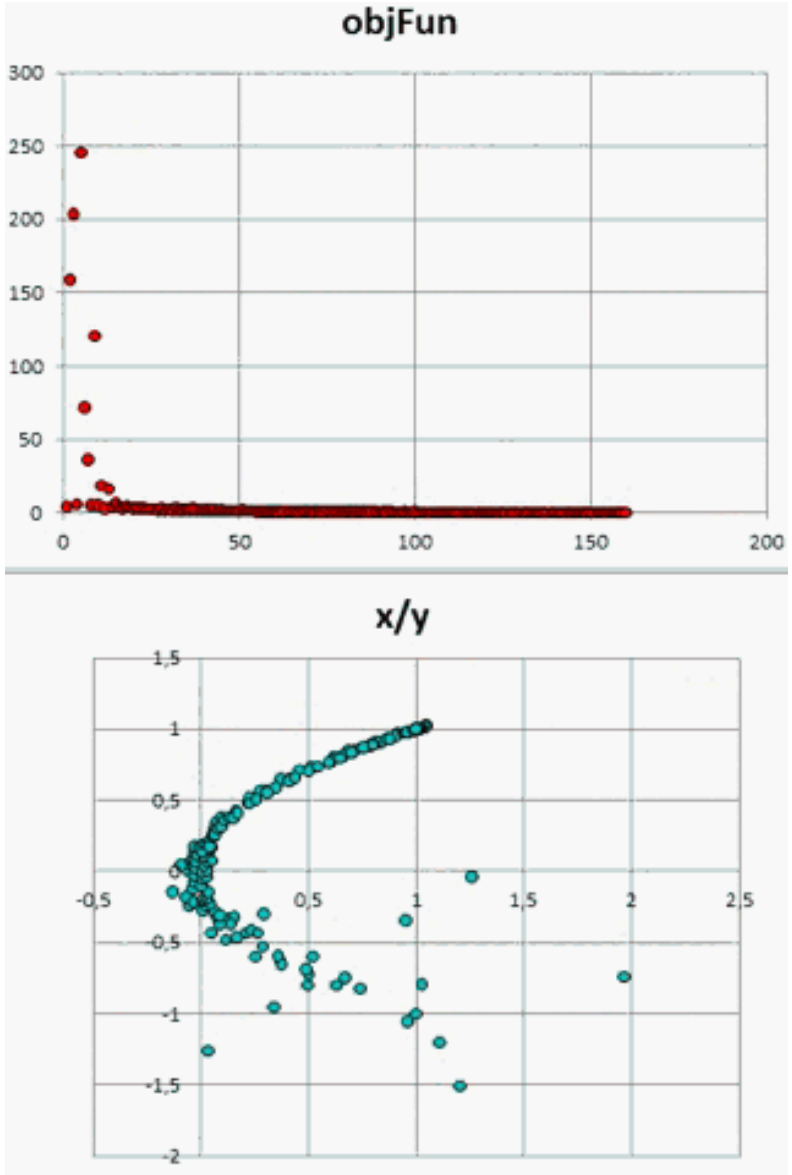
Nelder-Mead algoritmus



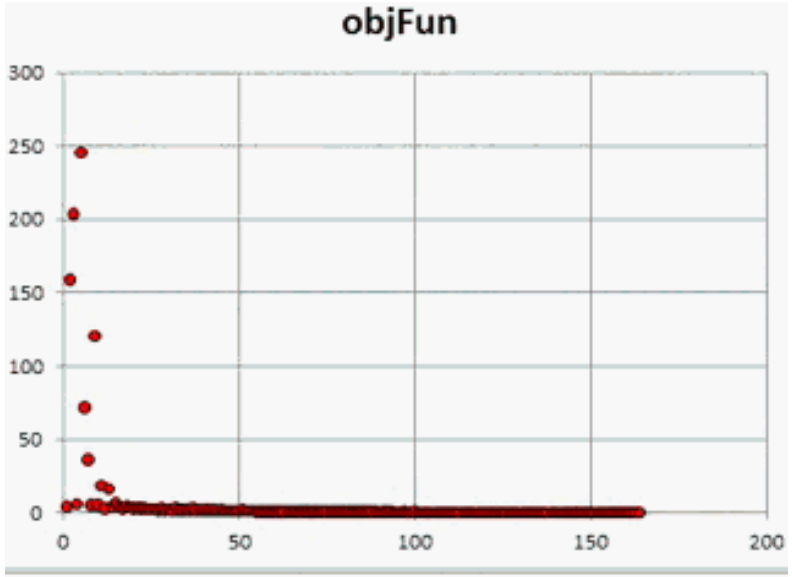
Nelder-Mead algoritmus

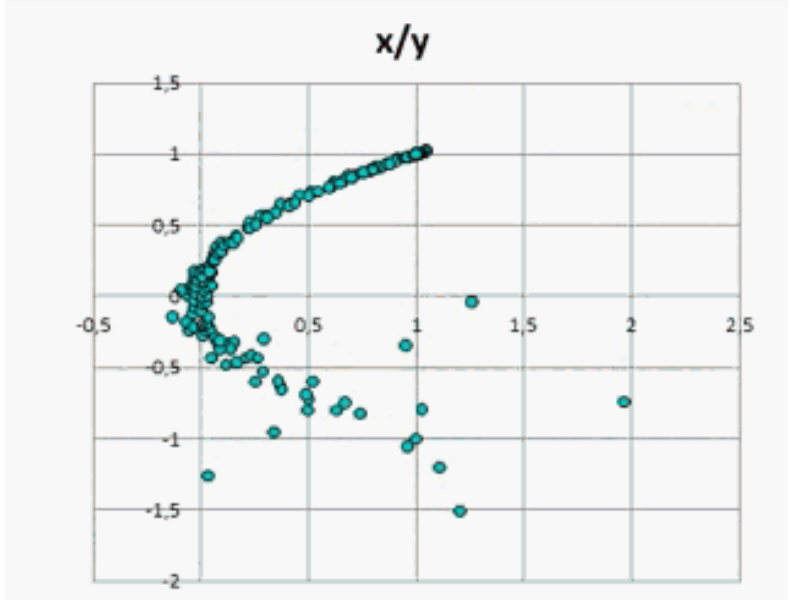


Nelder-Mead algoritmus

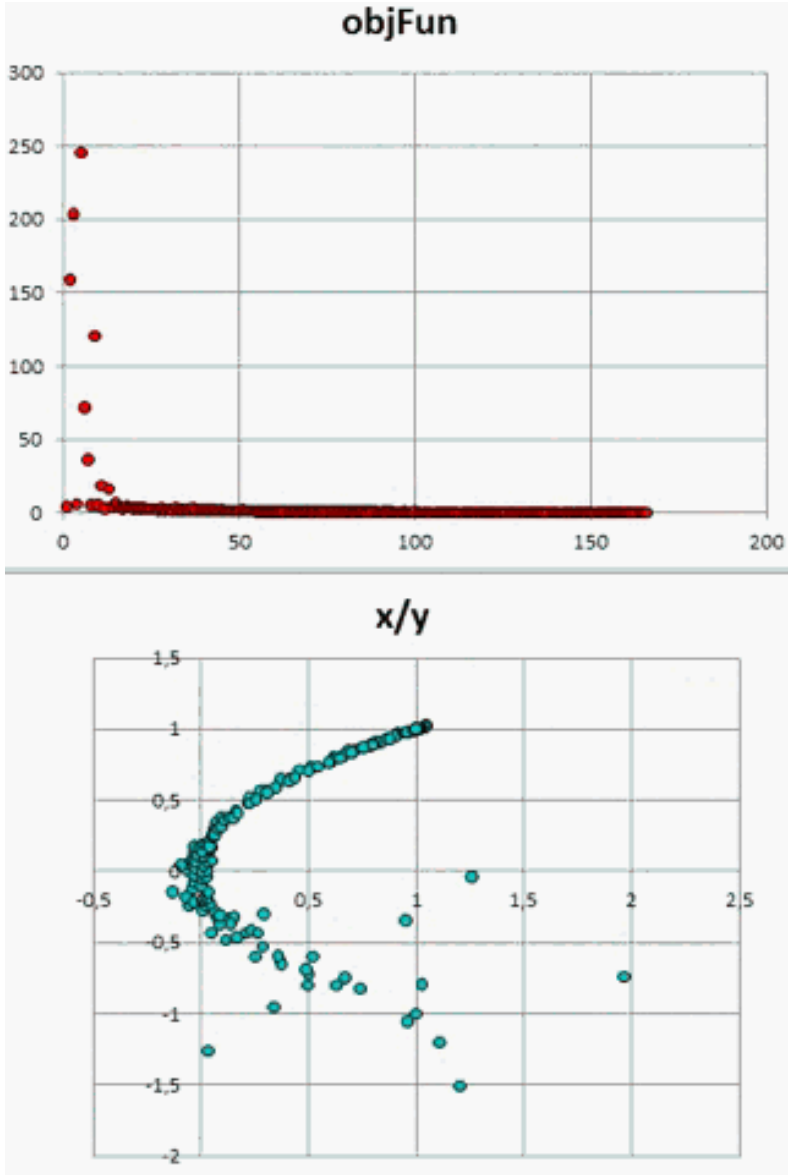


Nelder-Mead algoritmus

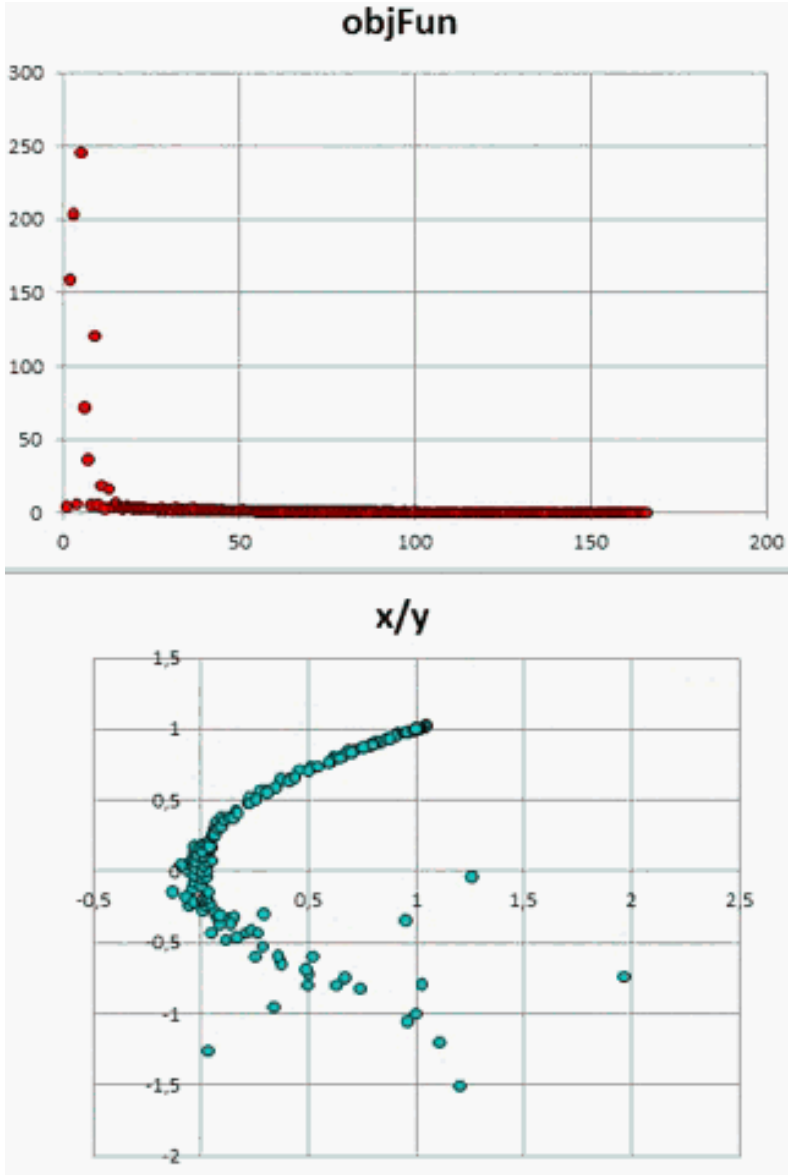




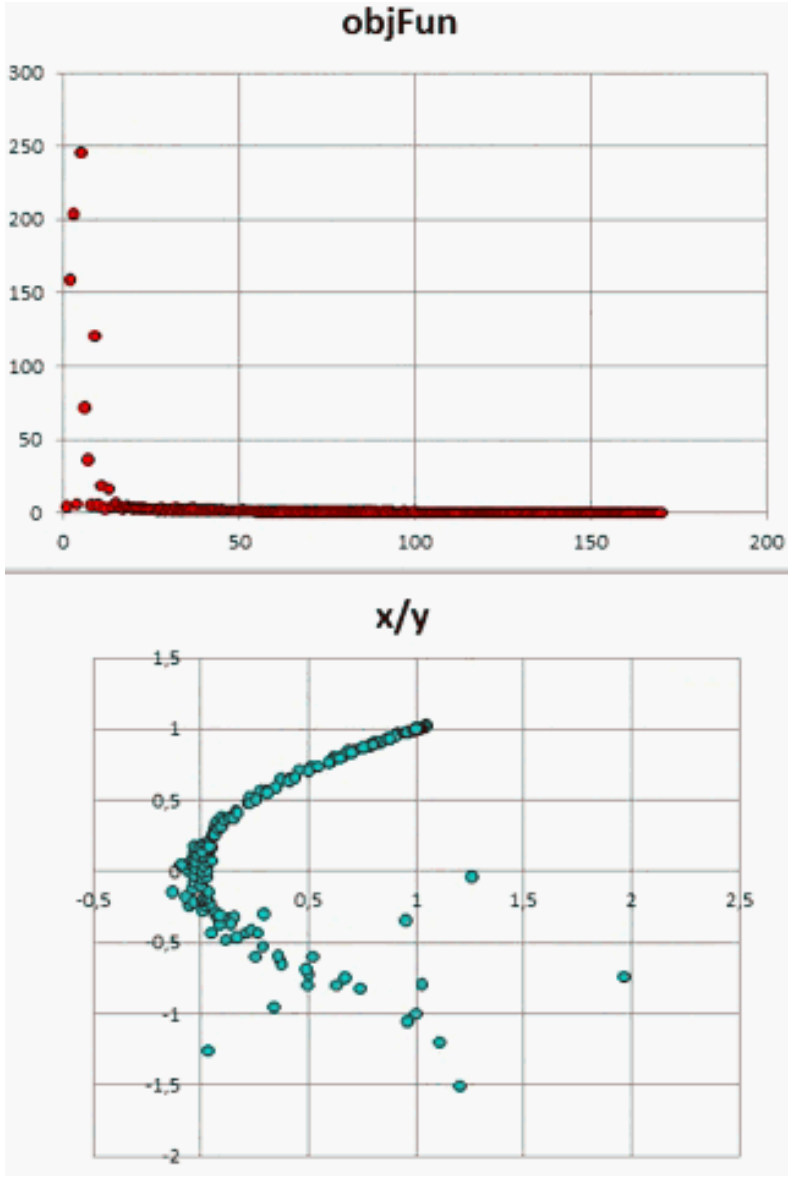
Nelder-Mead algoritmus



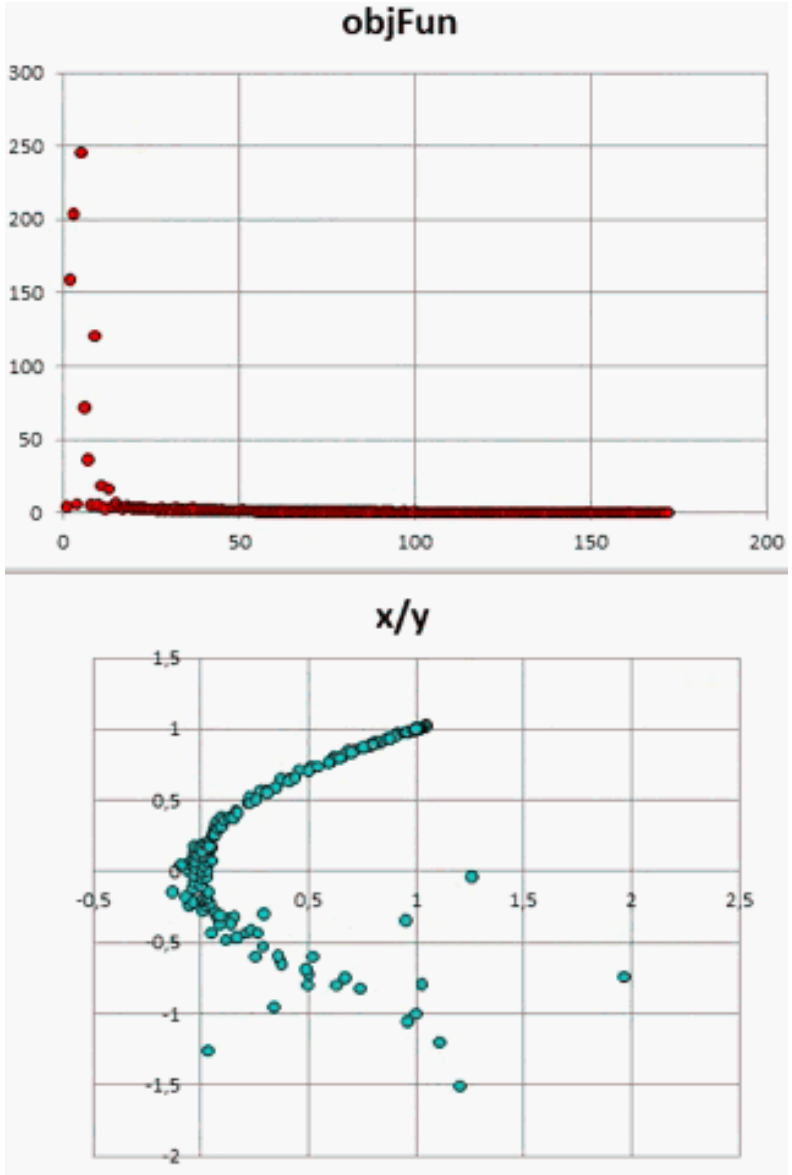
Nelder-Mead algoritmus



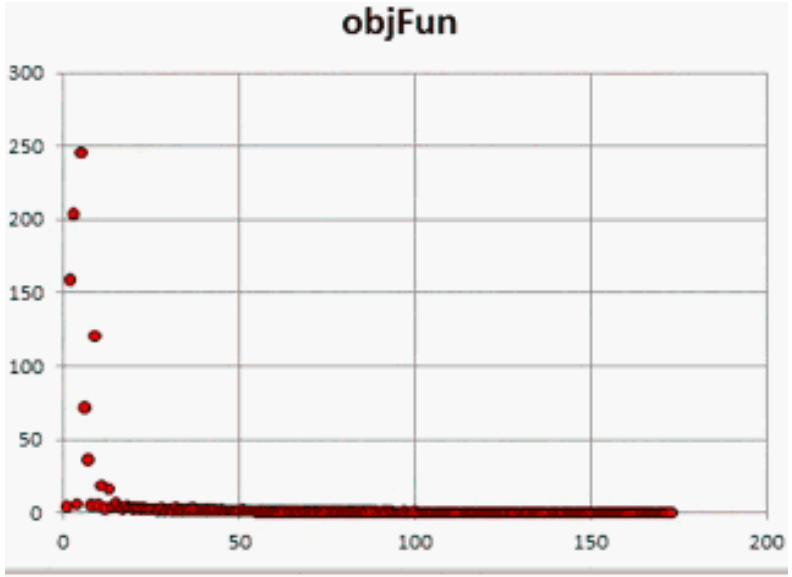
Nelder-Mead algoritmus

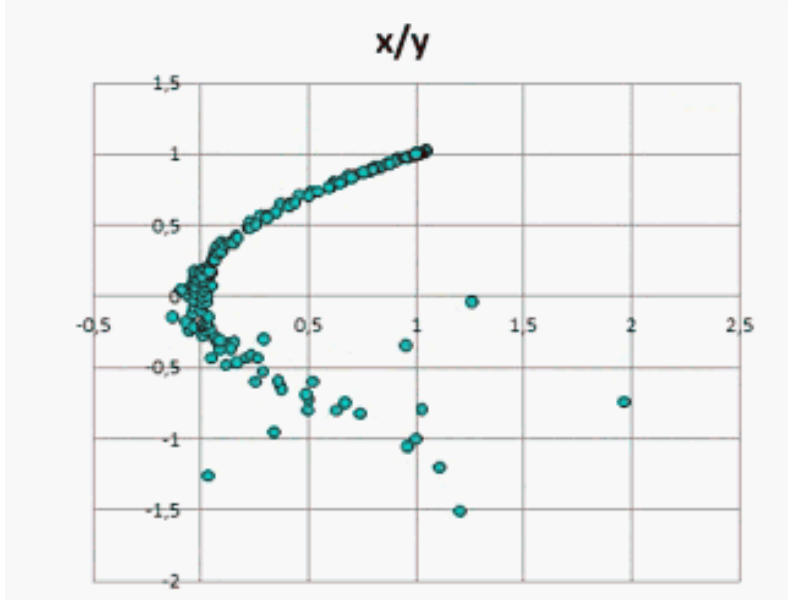


Nelder-Mead algoritmus

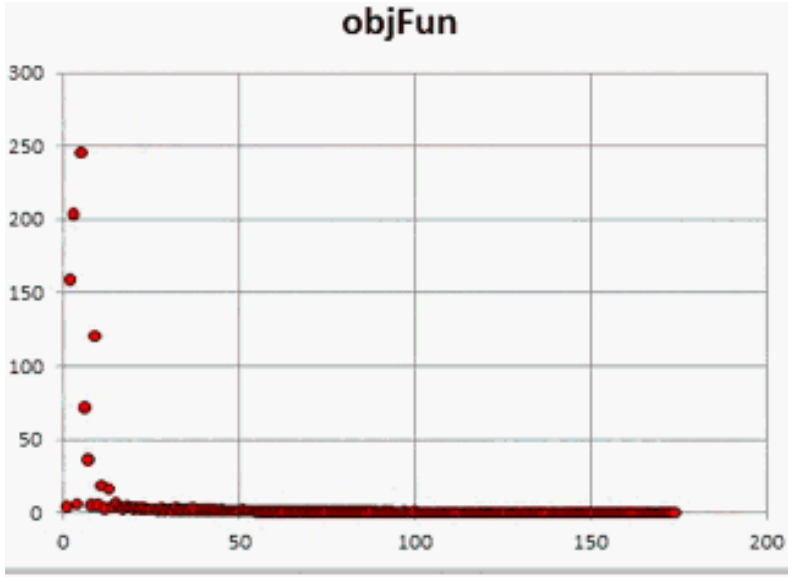


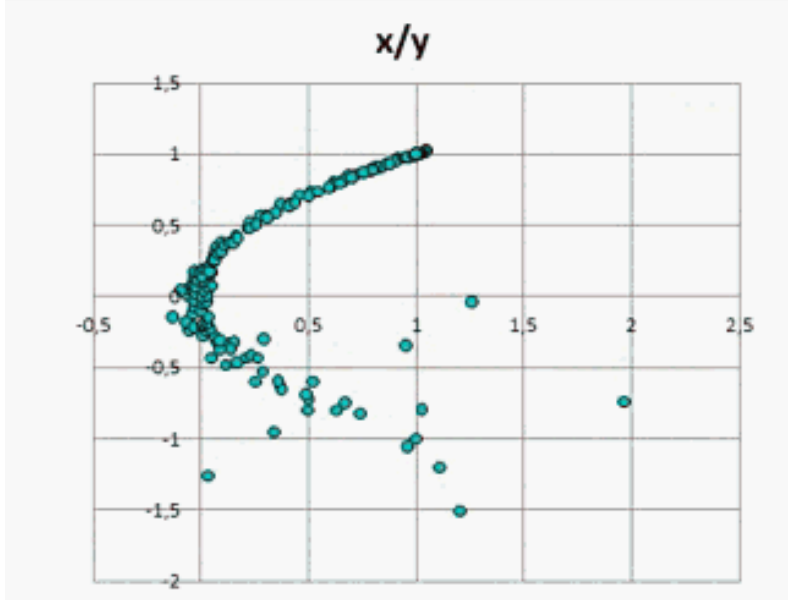
Nelder-Mead algoritmus



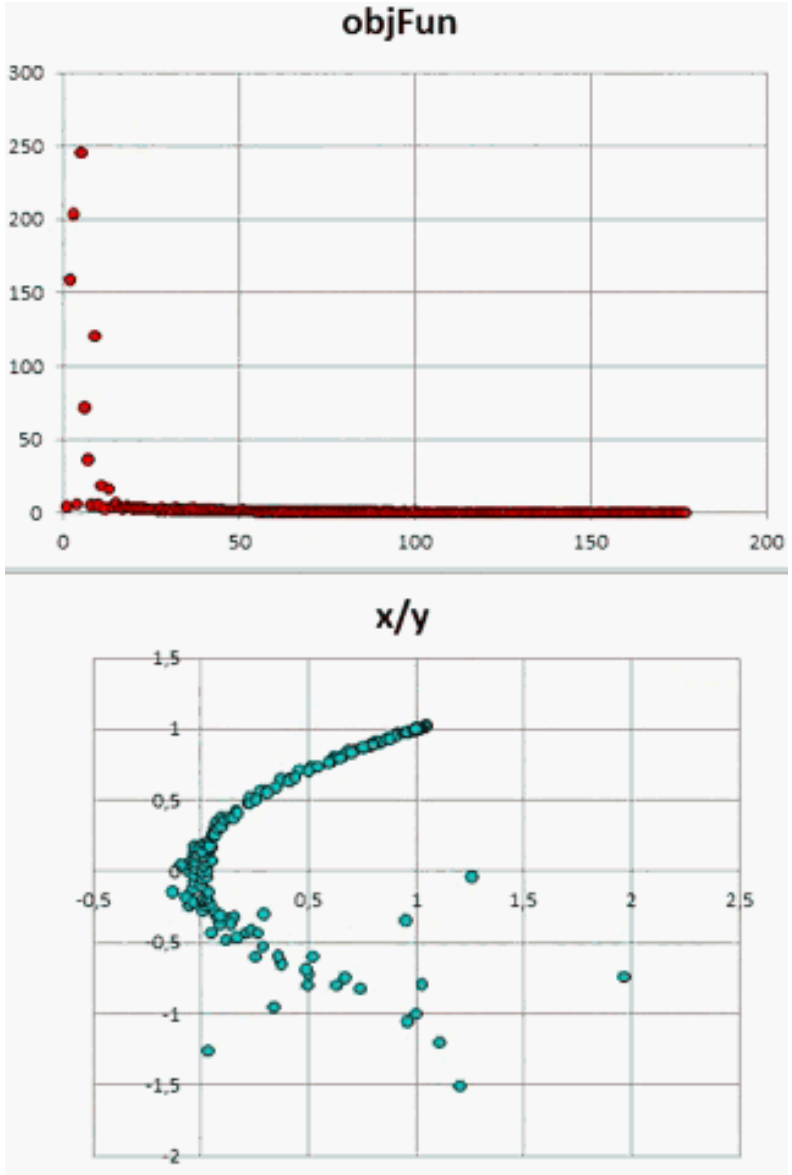


Nelder-Mead algoritmus

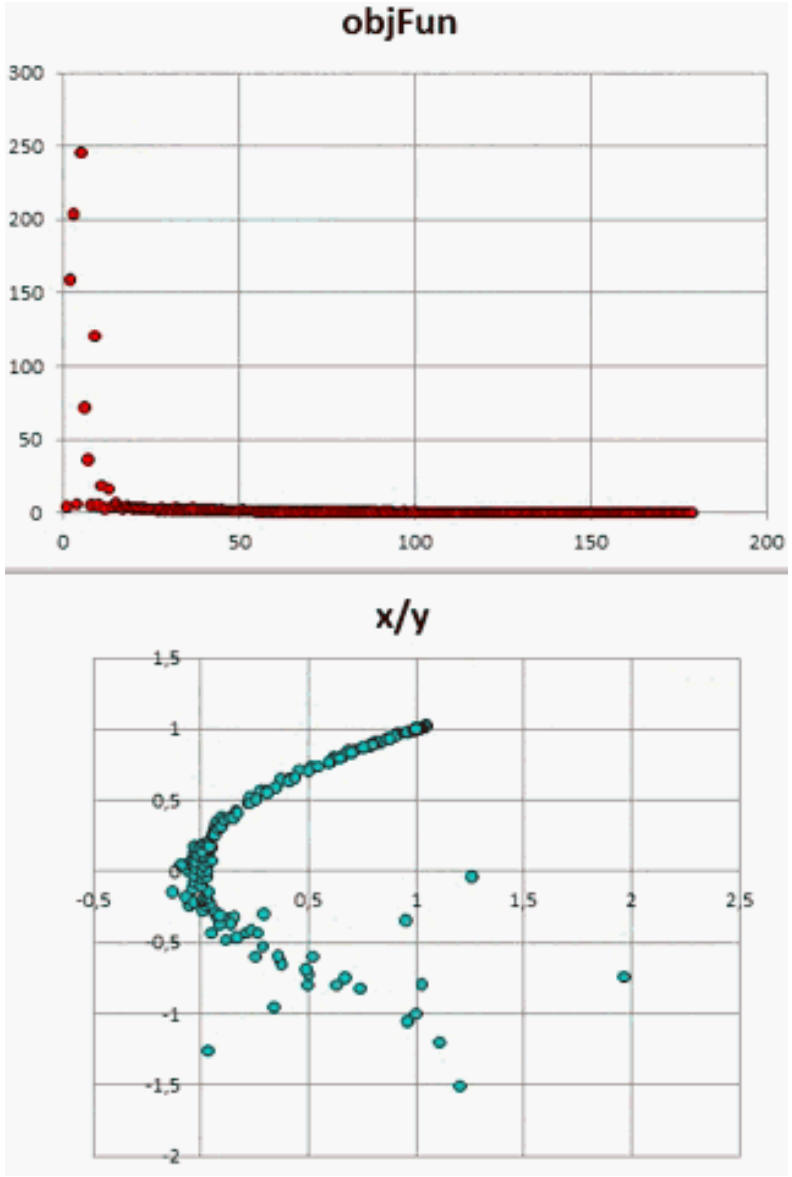




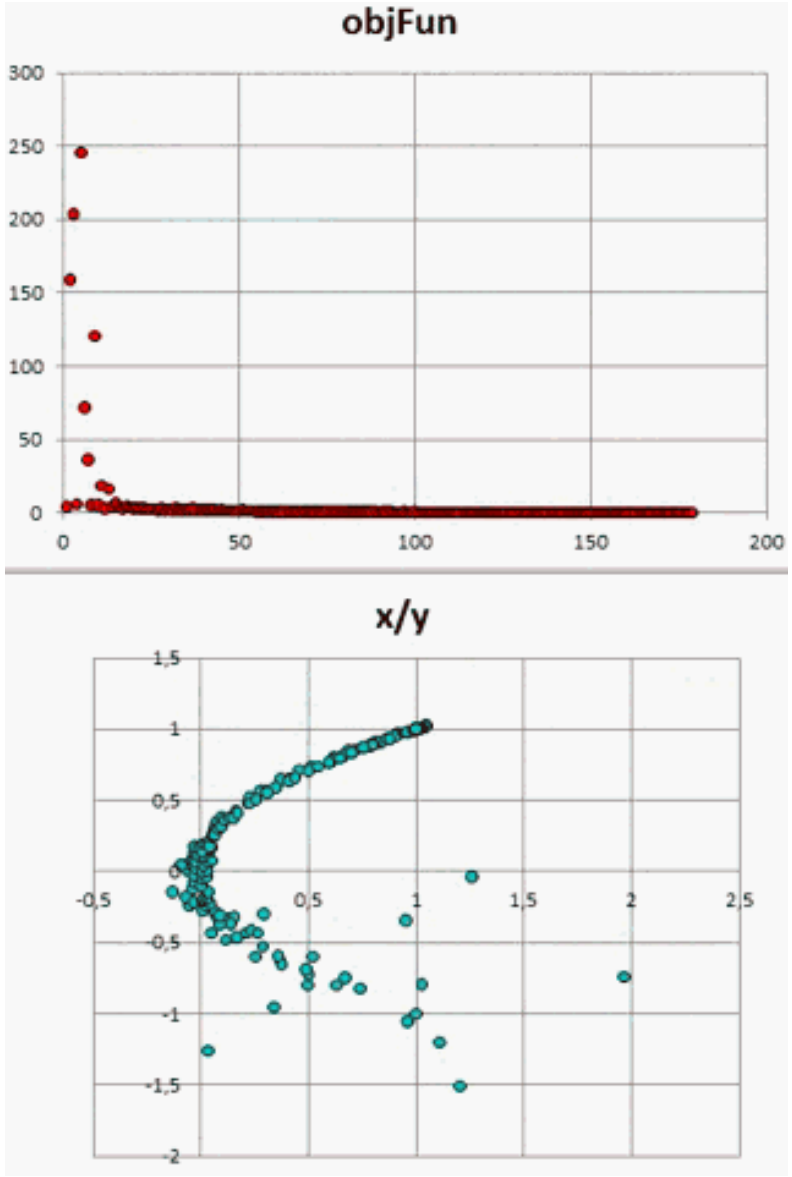
Nelder-Mead algoritmus



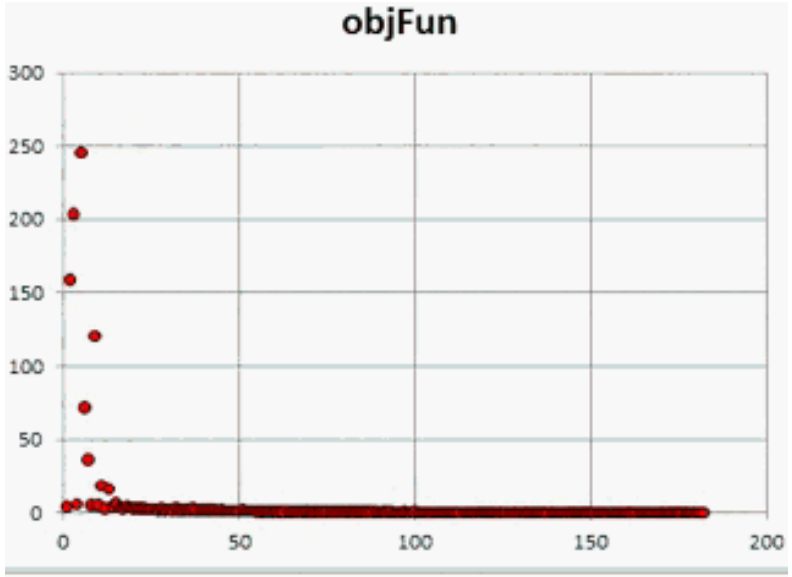
Nelder-Mead algoritmus

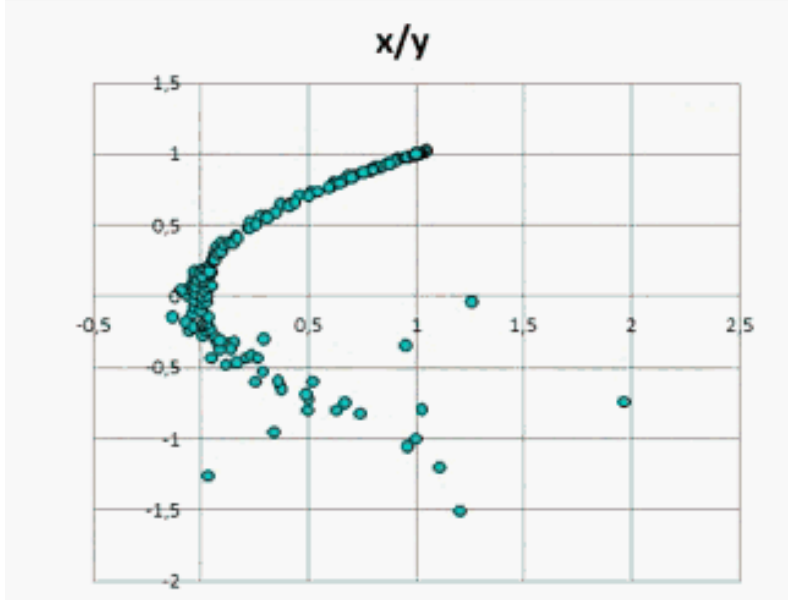


Nelder-Mead algoritmus

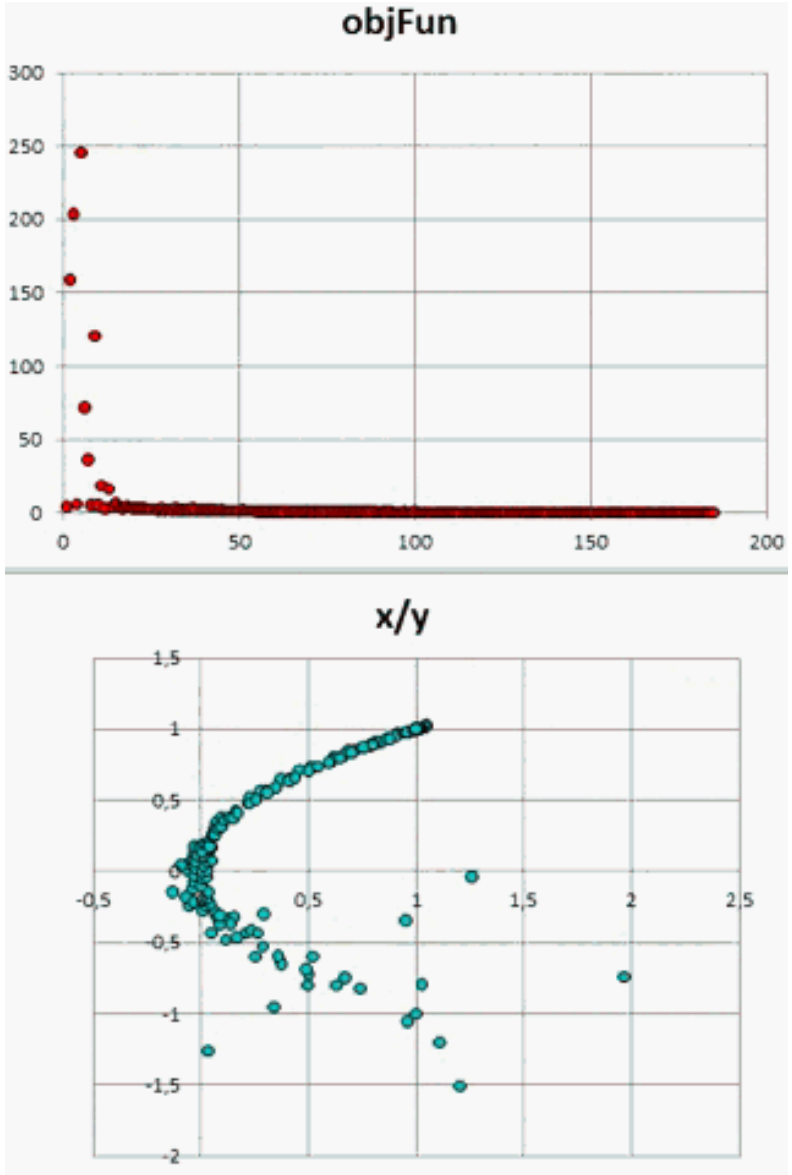


Nelder-Mead algoritmus

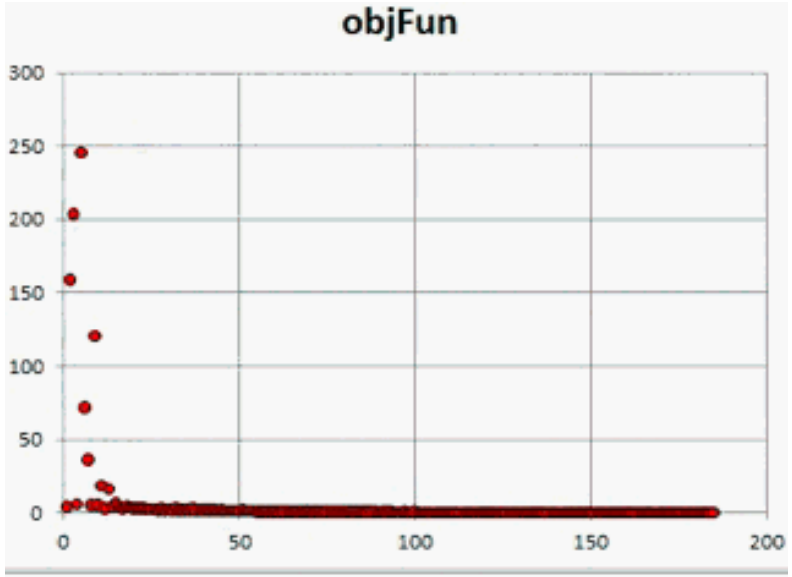


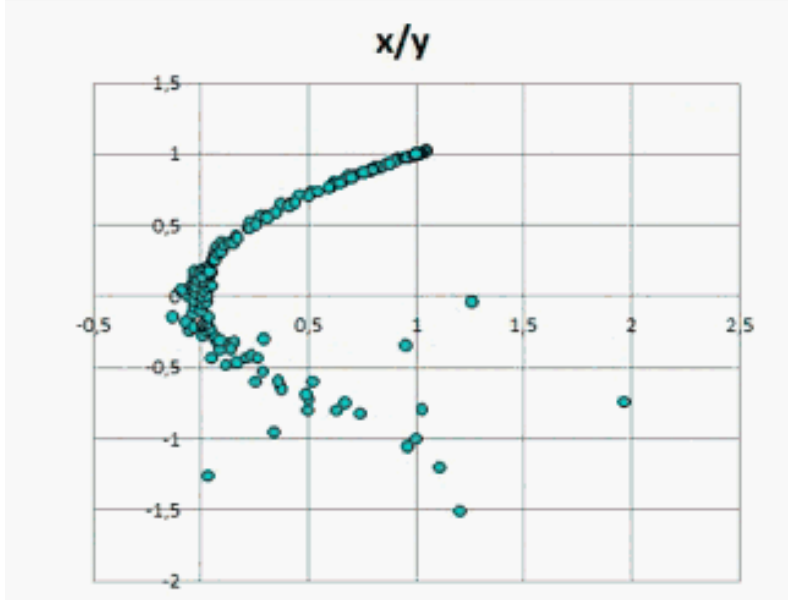


Nelder-Mead algoritmus



Nelder-Mead algoritmus





Nelder-Mead algoritmus

3. Feladat

Matlab-ban határozza meg az n-dimenziós Rosenbrock függvény függvény minimumát úgy, hogy a függvényt m-fájl alakjába definiálja. Tesztelje a Nelder-Mead algoritmust $n = 3, 4, 5$*-re.*

$$f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$$

Deriváltak $\mathbb{R}^n$-ben: ismétlés

Elsőrendű parciális derivált

Legyen $f : \mathbb{R}^n \to \mathbb{R}$. Az f függvény i-edik elsőrendű parciális deriváltja:

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_i + h, \ldots x_n) - f(x)}{h}.$$

Parciális deriváltnak nevezzük a többváltós függvények olyan deriváltját, amikor a függvényt egyik változójának függvényeként fogjuk fel, eszerint deriválunk, miközben a többi változót állandónak tekintjük.

Deriváltak $\mathbb{R}^n$-ben: ismétlés

Elsőrendű parciális derivált

Legyen $f : \mathbb{R}^n \to \mathbb{R}$. Az f függvény i-edik elsőrendű parciális deriváltja:

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_i + h, \ldots x_n) - f(x)}{h}.$$

Parciális deriváltnak nevezzük a többváltós függvények olyan deriváltját, amikor a függvényt egyik változójának függvényeként fogjuk fel, eszerint deriválunk, miközben a többi változót állandónak tekintjük.

A függvény gradiense

Legyen $f : \mathbb{R}^n \to \mathbb{R}$ differenciálható függvény. Az f függvény $\nabla f(x)$ *gradiense* az elsőrendű pariális deriváltjaival adott:

$$\nabla f(x) = \left[\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \ldots, \frac{\partial f}{\partial x_n}(x)\right]$$

Deriváltak $\mathbb{R}^n$-ben: ismétlés

4. Feladat

Határozza meg a következő függvények elsőrendű parciális deriváltjait és a gradiensét:

a) $f(x, y) = x^2 y$;

b) $f(x, y) = 4xy + \ln xy$;

Deriváltak $\mathbb{R}^n$-ben: ismétlés

Másodrendű parciális deriváltak

A $z = f(x, y)$ kétváltozós függvény másodrendű parciális deriváltjai:

- az x szerinti másodrendű parciális derivált:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right);$$

- az y szerinti másodrendű parciális derivált:

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial y}\right);$$

- a vegyes másodrendű parciális deriváltak:

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) \text{ és } \frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right).$$

Deriváltak $\mathbb{R}^n$-ben: ismétlés

A Hesse mátrix

A $H(x)$ mátrixot Hesse mátrixnak nevezzük, elemei az f függvény másodrendű parciális deriváltjai, azaz

$$H(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{bmatrix}$$

Deriváltak $\mathbb{R}^n$-ben: ismétlés

5. Feladat

Határozza meg az $f(x,y) = 3x^2y^3 - 7x^3y^3 + 10x^2 - 3x + 8y^h - 4y + 12$ *függvény gradiensét és Hesse mátrixát.*

Deriváltak $\mathbb{R}^n$-ben: ismétlés

5. Feladat

Határozza meg az $f(x,y) = 3x^2y^3 - 7x^3y^3 + 10x^2 - 3x + 8y^h - 4y + 12$ *függvény gradiensét és Hesse mátrixát.*

Deriváltak numerikus számítása Matlab-ban

Nincsen beépített függvény, viszont lehet programozni:

- gradiens számolása egy pontban: NumJacob.m függvény segítségével;
- Hesse mátrix számolása egy pontban: NumHessian.m függvény segítségével;

Deriváltak $\mathbb{R}^n$-ben: ismétlés

6. Feladat

Határozd meg az $f(x_1, x_2) = x_1^2 \cdot x_2^3$ *függvény gradiensét és Hesse mátrixát az* $(1, 2)$ *pontban. A feladatot old meg Matlab-ban is.*

Hogyan ismerjük fel a lokális minimumot?

Tétel (Elsőrendű szükséges feltétel)

Legyen $x^* \in \mathbb{R}^n$ az $f : \mathbb{R}^n \to \mathbb{R}$ folytonosan deriválható függvény lokális minimuma az x^* pont egy nyílt környezetében. Akkor $\nabla f(x^*) = 0$.

Stacionárius pont

Az x^* pontot, melyre érvényes $\nabla f(x^*) = 0$, a függvény stacionárius pontjának nevezzük. Ha a függvénynek van szélsőértéke, akkor biztosan a stacionárius pontban lesz.

Hogyan ismerjük fel a lokális minimumot?

Tétel (Másodrendű szükséges feltétel)

Tegyük fel, hogy $x^* \in \mathbb{R}^n$ az $f : \mathbb{R}^n \to \mathbb{R}$ függvény lokális minimuma és létezik a $H(x)$ Hesse mátrix és folytonos az x^* nyílt környezetében. Akkor $\nabla f(x^*) = 0$ és $H(x^*)$ pozitív szemidefinit.

Mátrixok definitsége

Legyen A egy kvadratikus mátrix. Akkor az A mátrix:

- pozitív definit, ha minden $x \in \mathbb{R}^n$, $x \neq 0$ vektorra $x^T Ax > 0$.
- pozitív szemidefinit, ha minden $x \in \mathbb{R}^n$, $x \neq 0$ vektorra $x^T Ax \geq 0$.
- negatív definit, ha minden $x \in \mathbb{R}^n$, $x \neq 0$ vektorra $x^T Ax < 0$.
- negatív szemidefinit, ha minden $x \in \mathbb{R}^n$, $x \neq 0$ vektorra $x^T Ax \leq 0$.

A fenti kategóriákba nem sorolható kvadratikus mátrixokat indefinit mátrixoknak nevezzük.

Hogyan ismerjük fel a lokális minimumot?

Tétel (Másodrendű elégséges feltétel)

Tegyük fel, hogy az $f : \mathbb{R}^n \to \mathbb{R}$ függvény $H(x)$ Hesse mátrixa folytonos az $x^* \in \mathbb{R}^n$ pont egy nyílt környezetében és $\nabla f(x^*) = 0$ és $H(x^*)$ pozitív definit. Akkor x^* a függvény szigorú lokális minimuma.

Hogyan ismerjük fel a lokális minimumot?

Tétel (Másodrendű elégséges feltétel)

Tegyük fel, hogy az $f : \mathbb{R}^n \to \mathbb{R}$ függvény $H(x)$ Hesse mátrixa folytonos az $x^* \in \mathbb{R}^n$ pont egy nyílt környezetében és $\nabla f(x^*) = 0$ és $H(x^*)$ pozitív definit. Akkor x^* a függvény szigorú lokális minimuma.

Tétel (Konvex függvény)

Az $f : \mathbb{R}^n \to \mathbb{R}$ konvex függvény x^* lokális minimumai egyben a globális minimumai is. Továbbá, ha a függvény deriválható, akkor az függvény mindegyik x^* stacionárius pontja egyben a függvény globális minimuma is.

Az optimumot kereső eljárásokról

1. mindegyik eljárás egy x_0 kezdőpontot követel;
2. az eljárás egy $\{x_k\}_{k\in\mathbb{N}}$ végtelen sorozatot generálnak;
3. az eljárás felhasználja a függvény által nyújtott információkat, hogy az $x_0, x_1, \ldots, x_k$ pont alapján eldöntse, hogyan határozza meg az x_{k+1} pontot;

Az optimumot kereső eljárásokról

1. mindegyik eljárás egy x_0 kezdőpontot követel;
2. az eljárás egy $\{x_k\}_{k\in\mathbb{N}}$ végtelen sorozatot generálnak;
3. az eljárás felhasználja a függvény által nyújtott információkat, hogy az $x_0, x_1, \ldots, x_k$ pont alapján eldöntse, hogyan határozza meg az x_{k+1} pontot;

A deriváltaka használó eljárások általában a következő stratégiákat használják az x_{k+1} pont meghatározására:

- vonalmenti minimalizálás módszere;
- trust-region módszere;

Vonalmenti minimaliázálás

Olyan stratégia, amely megmondja, hogyan határozzuk meg az x_{k+1} pontot, ha ismerjük az x_k pontot.

Vonalmenti minimaliázálás

Olyan stratégia, amely megmondja, hogyan határozzuk meg az x_{k+1} pontot, ha ismerjük az x_k pontot.
Egy olyan p_k kutatási irányt határoz meg, amely irányában keressük azt az x_{k+1} pontot, melyben a célfüggvény értéke kisebb lesz az $f(x_k)$-nál. Ezt a kutatási irányt a

$$\min_{\alpha>0} f(x_k + \alpha p_k)$$

egydimenziós minimumkeresési probléma megoldásával határozzuk meg, amit az α lépéshossz szerint oldunk meg.
A fenti minimumkeresési problémát szükségtelen pontosan megoldani. A vonalmenti minimalizálás algoritmusa korlátolt számú lépéshosszt generál addig, még rátalát a probléma közelítő megoldására.

Vonalmenti minimalizálás algoritmusa

1. lépés: Bemenetel: $x_0 \in \mathbb{R}^n$, $k = 1$.
2. lépés: Válasszuk meg a $p_k \in \mathbb{R}^n$ kutatási irányt.
3. lépés: Válasszuk meg az $\alpha_k > 0$ lépéshosszt a

$$\min_{\alpha > 0} f(x_k + \alpha p_k)$$

probléma megoldásával.

4. lépés: $x_{k+1} = x_k + \alpha_k p_k$.
5. lépés: Legyen $k = k + 1$ és menj a 2. lépésre.

Kutatási irány

A kutatási iránynak csökkenőnek kell lennie, vagyis

$$p_k^T \nabla f_k < 0,$$

ami biztosítja, hogy

$$f(x_0) \geq f(x_1) \geq f(x_2) \geq \ldots \geq f(x_k) \geq f(x_{k+1}) \geq \ldots.$$

A kutatási iránynak általában a következő alakja van

$$p_k = -B_k^{-1} \nabla f_k.$$

A B_k mátrixtól függően különböző módszerek léteznek.

Kutatási irány

A kutatási iránynak csökkenőnek kell lennie, vagyis

$$p_k^T \nabla f_k < 0,$$

ami biztosítja, hogy

$$f(x_0) \geq f(x_1) \geq f(x_2) \geq \ldots \geq f(x_k) \geq f(x_{k+1}) \geq \ldots.$$

A kutatási iránynak általában a következő alakja van

$$p_k = -B_k^{-1} \nabla f_k.$$

A B_k mátrixtól függően különböző módszerek léteznek.

A leggyorsabb lecsökkenés módszere

$B_k = I$, ahol I egységmátrix, vagyis a kutatási irány:

$$p_k = -\nabla f_k.$$

Kutatási irány

Newton-szerű módszerek

A B_k szimmetrikus pozitív definit mátrix.

Kutatási irány

Newton-szerű módszerek

A B_k szimmetrikus pozitív definit mátrix.

Newton módszer: $B_k = H(x_k)$, ahol $H(x_k)$ a célfüggvény Hesse mátrixának értéke az x_k pontban.

Kutatási irány

Newton-szerű módszerek

A B_k szimmetrikus pozitív definit mátrix.

Newton módszer: $B_k = H(x_k)$, ahol $H(x_k)$ a célfüggvény Hesse mátrixának értéke az x_k pontban.

Kvázi-Newton módszerek: a Hesse mátrixot approximálják bizonyos képlet segítségével.

Kutatási irány

Kvázi-Newton módszerek

Davidon-Fletcher-Powell (DFP) eljárás:

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k},$$

ahol $s_k = x_{k+1} - x_k = -H_k \nabla f_k$, $y_k = \nabla f_{k+1} - \nabla f_k$. Vegyük észre, hogy $H_k = B_k^{-1}$.

Kutatási irány

Kvázi-Newton módszerek

Davidon-Fletcher-Powell (DFP) eljárás:

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k},$$

ahol $s_k = x_{k+1} - x_k = -H_k \nabla f_k$, $y_k = \nabla f_{k+1} - \nabla f_k$. Vegyük észre, hogy $H_k = B_k^{-1}$.

Broyden-Fletcher-Goldfarb-Shanno (BFGS):

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I = \rho_k y_k s_k^T) + \rho_k s_k s_k^T,$$

ahol $\rho_k = \frac{1}{y_k^T s_k}$. Általában $H_0 = I$.

Lépéshossz

Az α_k lépéshossz meghatározásakor kompromisszumot kell kötni: a lépéshossz a célfüggvény elégséges csökkenését kell eredményeznie, viszont nem szeretnénk túl sok időt és energiát "pazarolni" a

$$\min_{\alpha>0} f(x_k + \alpha p_k)$$

probléma megoldására. Az ilyen lépéshossznak eleget kell tennie bizonyos feltételeknek

Lépéshossz

Armijo feltétel

Biztosítja a célfüggvény elfogadható mértékű csökkenését:

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k,$$

ahol $c_1 \in (0, 1)$. Általában $c_1 = 10^{-4}$. Minden elég kicsi $\alpha > 0$ eleget tesz az Armijo feltételnek, ezért kell még egy feltétel, amely a lépéshosszat alulról korlátozza.

Lépéshossz

Armijo feltétel

Biztosítja a célfüggvény elfogadható mértékű csökkenését:

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k,$$

ahol $c_1 \in (0, 1)$. Általában $c_1 = 10^{-4}$. Minden elég kicsi $\alpha > 0$ eleget tesz az Armijo feltételnek, ezért kell még egy feltétel, amely a lépéshosszat alulról korlátozza.

Wolfe feltétel

Kiszűri a túl rövid lépéshosszakat.

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k,$$

ahol $c_2 \in (c_1, 1)$. Általában $c_2 = 0.9$ a Newton-szerű módszereknél.

Lépéshossz

Backtracking algoritmus

A lépéshossz meghatározásánál elhanyagolhatjuk a Wolfe feltételt, ha a következő algoritmust használjuk:

1. lépés: válassz $\bar{\alpha} > 0, \rho, c \in (0,1)$, legyen $\alpha = \bar{\alpha}$;
2. lépés: ellenőrízd le, teljesül-e az Armijo feltétel;
3. lépés: ha nem teljesül, $\alpha = \rho\alpha$, menj vissza a 2. lépésre, különben $\alpha_k = \alpha$, STOP.

A Newton-szerű módszereknél $\bar{\alpha} = 1$ és általában $\rho = \frac{1}{2}$.

Trust-region módszerek

Olyan stratégia, amely a függvény által hordozott információkat használja fel arra, hogy egy m_k modellfüggvényt szerkesszen, amely az x_k környezetében jól approximálja a függvényt. Ezt a környezetet "trust region"-nek, vagyis megbízhatósági tartománynak nevezzük. Ebben a környezetben keressük az m_k modellfüggvény minimumát, vagyis a következő minimalizálási problémát kell megoldani a p lépés szerint:

$$\min_p m_k(x_k + p), \quad \text{ahol } x_k + p \text{ a megbízhatósági tartományban fekszik.}$$

Ha a fenti probléma megoldása nem hozza meg az f függvény elégséges csökkenését, akkor csökkentjük a megbízhatósági tartományt és újra megoldjuk a fenti problémát.

Trust-region módszerek

A modellfüggvény

Az $f(x)$ célfüggvényt másodrendű Taylor polinommal approximáljuk az x_k iteráció környezetében:

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p,$$

ahol B_k valamilyen szimmetrikus mátrix, p a lépés.

Trust-region módszerek

A modellfüggvény

Az $f(x)$ célfüggvényt másodrendű Taylor polinommal approximáljuk az x_k iteráció környezetében:

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p,$$

ahol B_k valamilyen szimmetrikus mátrix, p a lépés.

A p lépés meghatározása

A következő feltételes optimalizálási feladatot (nemlineáris programozási feladatot) kell megoldani:

$$\min_{p \in \mathbb{R}^n} m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p, \text{ f.h. } \|p\| \leq \triangle_k,$$

ahol $\triangle_k > 0$ a megbízhatósági tartomány sugara.

Trust-region módszerek

A p lépés meghatározása

- ha $\triangle_k$ elég nagy, a fenti probléma feltétel nélküli optimalizációs feladatra vezetődik vissza és a lépést teljes lépésnek nevezzük: $p_k^B = -B_k^{-1}\nabla f_k$;
- ha $\triangle_k$ elég kicsi, akkor feltételes optimilizási feladat, ahol a célfüggvény és a megszorítás is kvadratikus;

Trust-region módszerek

A megbízhatósági tartomány sugarának meghatározása

A $\triangle_k$ meghatározásában fontos szerepet játszik a

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} = \frac{\text{a célfüggvény tényleges csökkenése}}{\text{a célfüggvény jelzett csökkenése}}.$$

Ha

- $\rho_k < 0$, vagy a $\rho_k \approx 0$ a megbízhatósági tartományt szűkítjük, vagyis csökkentjük a $\triangle_k$-t;
- $\rho \approx 1$, a megbízhatósági tartományt bővítjük, vagyis a $\triangle_k$-t növeljük;
- $\rho > 0$, de nincs közel 1-hez és 0-hoz sem, nem változtatjuk a megbízhatósági tartományt;

Trust-region algoritmus I

Bemenetel: Kezdő iteráció $x_0 \in \mathbb{R}^n$, konstansok $\bar{\triangle} > 0, \triangle_0 \in (0, \bar{\triangle})$ és $\eta \in \left[0, \frac{1}{4}\right)$;

$k \leftarrow 0$;

while a megállási feltétel nem teljesül

  1. Lépés (Lépés meghatározása);

  Megközelítően old meg a p szerint a feltételes minimalizálási feladatot:

  $\min_{p \in \mathbb{R}^n} m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p$, f.h. $\|p\| \leq \triangle_k$.

  2. Lépés (Megbízhatósági tartomány frissítése)

  Számold ki a: $\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$.

  if $\rho_k < \frac{1}{4}$

    $\triangle_{k+1} = \frac{1}{4}\|p_k\|$;

  else

    if $\rho_k > \frac{3}{4}$ és $\|p_k\| = \triangle_k$

Trust-region algoritmus II

```
            △_{k+1} = min{2△_k, △̄};
        else
            △_{k+1} = △_k;
        end if
    end if
    if ρ_k > η
        x_{k+1} = x_k + p_k;
    else
        x_{k+1} = x_k;
    end if
    k ← k + 1;
end while
```

A minimalizálási részfeladat megoldása

Cauchy pont

Az a pont ami a célfüggvény lehető legkisebb mértékű csökkenését eredményezi:

$$p_k^c = -\tau_k \frac{\triangle_k}{\|\nabla f_k\|} \nabla f_k,$$

ahol

$$\tau_k = \begin{cases} 1 & \text{ha } \nabla f_k^T B_k \nabla f_{\leq} 0, \\ \min\{\frac{\|\nabla f_k\|^3}{\triangle_k \nabla f_k^T B_k \nabla f_k}, 1\} & \text{különben.} \end{cases}$$

A minimalizálási részfeladat megoldása

A dogleg módszer

A Cauchy pont feljavítására szolgál. Akkor használják, ha a B_k pozitív definit. A dogleg módszer szerint a p_k lépést a következő stratégia szerint határozhatjuk meg:

- Legyen $p_k^B = B_k^{-1}\nabla f_k$ a teljes lépés. Ha $\|p_k^B\| \leq \triangle_k$, akkor $p_k = p_k^B$.
- Ha $\triangle_k$ nagyon kicsi, akkor $p_k \approx -\triangle_k \frac{\nabla f_k}{\|\nabla f_k\|}$.

A minimalizálási részfeladat megoldása

A dogleg módszer

- Ha $\triangle_k$ se nem túl kicsi, se nem túl nagy, akkor a lépés approximációját meg lehet kapni mint az

$$\|p_k^U + (\tau_k - 1)(p_k^B - p_k^U)\|^2 = \triangle_k^2$$

egyenlet megoldását a $\tau > 0$ szerint. Az egyenletben ismert

$$p_k^U = -\frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T B_k \nabla f_k} \nabla f_k.$$

Ha meghatároztuk a τ_k értékét, akkor

$$p_k \approx \begin{cases} \tau_k p_k^U, & \text{ha } 0 \leq \tau_k \leq 1 \\ p_k^U + (\tau_k - 1)(p_k^B - p_k^U), & \text{ha } 1 \leq \tau_k \leq 2. \end{cases}$$

A B_k mátrix meghatározása

Az

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p$$

modellfüggvényben meg kell határozni a B_k szimmetrikus mátrixot. Attól függően, hogyan határozzuk meg a mátrixot, a trust-region módszer lehet:

A B_k mátrix meghatározása

Az

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p$$

modellfüggvényben meg kell határozni a B_k szimmetrikus mátrixot. Attól függően, hogyan határozzuk meg a mátrixot, a trust-region módszer lehet:

Newton trust-region módszer: $B_k = H(x_k)$, ahol $H(x_k)$ a célfüggvény Hesse mátrixának értéke az x_k pontban.

A B_k mátrix meghatározása

Az

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p$$

modellfüggvényben meg kell határozni a B_k szimmetrikus mátrixot. Attól függően, hogyan határozzuk meg a mátrixot, a trust-region módszer lehet:

Newton trust-region módszer: $B_k = H(x_k)$, ahol $H(x_k)$ a célfüggvény Hesse mátrixának értéke az x_k pontban.

Kvázi-Newton trust-region módszerek: a Hesse mátrixot approximálják bizonyos képlet segítségével.

A B_k mátrix meghatározása

Kvázi-Newton trust-region módszerek

Davidon-Fletcher-Powell (DFP) eljárás:

$$B_{k+1} = (I - \gamma_k y_k s_k^T) B_k (I - \gamma_k s_k y_k^T) + \gamma_k y_k y_k^T,$$

ahol $s_k = x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$, $\gamma_k = \frac{1}{y_k^T s_k}$.

A B_k mátrix meghatározása

Kvázi-Newton trust-region módszerek

Davidon-Fletcher-Powell (DFP) eljárás:

$$B_{k+1} = (I - \gamma_k y_k s_k^T) B_k (I - \gamma_k s_k y_k^T) + \gamma_k y_k y_k^T,$$

ahol $s_k = x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$, $\gamma_k = \frac{1}{y_k^T s_k}$.

Broyden-Fletcher-Goldfarb-Shanno (BFGS):

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}.$$

Feltétel nélküli minimalizálás a Matlab-ban

A fminunc függvény segítségével tudjuk alkalmazni vonalmenti minimalizálás legyorsabb lecsökkenés és kvázi-Newton módszerét, illetve a trust-region módszerét.

Feltétel nélküli minimalizálás a Matlab-ban

A fminunc függvény segítségével tudjuk alkalmazni vonalmenti minimalizálás legyorsabb lecsökkenés és kvázi-Newton módszerét, illetve a trust-region módszerét.

7. Feladat

Matlab segítségével határozza meg az

$$f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

függvény minimumát. A függvényt definiálja m-fájlba. A kezdő iteráció legyen $x_0 = [-1, 1]$. *Hasonlítsa össze a legyorsabb lecsökkenés és kvázi-Newton módszereit illetve a trust-region módszert a feladaton.*

A konvergencia sebesség

Az iteratív algoritmusok elmélete három részre osztható. Az első foglalkozik az algoritmusok építésével, figyelembe véve a feladat struktúráját. A második rész fő kérdése a módszerek konvergenciája, még pontosabban a globális konvergencia vizsgálata, aminek megléte azt jelenti, hogy a módszer által generált pontsorozat a megengedett tartomány tetszőleges pontjából indulva tart (konvergál) egy

a) stacionárius ponthoz;

b) lokális minimumhoz;

c) globális minimumhoz.

Az elmélet harmadik része a lokális konvergencia vizsgálatával foglalkozik, ami a konvergencia sebesség vizsgálatát jelenti.

A konvergencia sebesség

Definíció

Legyen $x^* \in \mathbb{R}^n$ *és* $\{x_k\}$ *olyan sorozat, melyre érvényes* $\lim_{k\to\infty} x_k = x$.

- *Az* $\{x_k\}$ *sorozat az* x^* *pont felé konvergál lineáris sebességgel, ha létezik olyan* $r \in (0,1)$, *melyre minden elég nagy* k *indexre érvényes*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq r.$$

- *A konvergencia szuperlineáris, ha* $\lim_{k\to\infty} \frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} = 0$.
- *A konvergencia kvadratikus, ha minden elég nagy* k *indexre*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \leq M.$$

A vonalmenti minimalizálás konvergenciája

Tétel (Globális konvergencia)

Tételezzük fel, hogy a vonalmenti minimalizálás eljárásban a p_k kutatási irány Newton-szerű módszerrel lett meghatározva (csökkenő) és az α_k lépéshossz eleget tesz az Armijo-Wolfe feltételeknek. Tételezzük fel, hogy az f célfüggvény alulról korlátos és folytonosan deriválható egy $\mathcal{N}$ nyílt tartományban amely tartalmazza az $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$ nivóhalmazt. Tételezzük fel, hogy a ∇f gradiens Lipschitz folytonos az $\mathcal{N}$ tartományban. Tételezzük fel, hogy a Newton-szerű módszerben a B_k pozitív definitek és mátrixok kondíciószáma egyenletesen korlátosak, vagyis létezik olyan $M > 0$, hogy minden k indexre érvényes:

$$\|B_k\|\|B_k^{-1}\| \leq M.$$

Akkor $\lim_{k\to\infty} \|\nabla f_k\| = 0$, vagyis a Newton-szerű vonalmenti minimalizálás eljárása globálisan konvergens.

A vonalmenti minimalizálás konvergenciája

Tétel (Lokális konvergencia)

Legyen az $f : \mathbb{R}^n \to \mathbb{R}$ célfüggvény háromszor folytonosan deriválható. Tételezzük fel, hogy a vonalmenti minimalizálás eljárásban a p_k kutatási irány kvázi-Newton módszerrel lett meghatározva és az α_k lépéshossz eleget tesz az Armijo-Wolfe feltételeknek, ahol $c_1 \leq \frac{1}{2}$. Ha az $\{x_k\}$ iterációs sorozat az x^* pont felé konvergál, melyre $\nabla f(x^*) = 0$ és $H(x^*)$ pozitív definit és érvényes:

$$\lim_{k\to\infty} \frac{(B_k - H(x^*))p_k\|}{\|p_k\|} = 0,$$

akkor

- *elég sok iteráció után $\alpha_k = 1$ minden k indexre;*
- *ha bekövetkezik az előző eset, $\{x_k\}$ szuperlineárisan konvergál x^* felé.*

A vonalmenti minimalizálás konvergenciája

Tétel (Newton-módszer lokális konvergenciája)

Legyen az $f : \mathbb{R}^n \to \mathbb{R}$ célfüggvény kétszer folytonosan deriválható és $H(x)$ Lipschitz folytonos az x^* környezetében, melyben $\nabla f(x^*) = 0$ és $H(x^*)$ pozitív definit. Tételezzük fel, hogy a kutatási irányt a Newton módszer segítségével határoztuk meg és $\alpha_k = 1$ minden k indexre. Akkor ha

1. *az x_0 kezdő iteráció elég közel van az x^* megoldáshoz, az $\{x_k\}$ iterációs sorozat az x^* megoldás felé konvergál;*
2. *a konvergencia kvadratikus;*
3. *a $\{\|\nabla f_k\|\}$ sorozat kvadratikusan konvergál a 0 felé.*

A trust-region eljárások konvergenciája

Tétel (Globális konvergencia)

A trust-region algoritmusban legyen $\eta \in \left(0, \frac{1}{4}\right)$. *Tegyük fel, hogy a függvénymodellben* $\|B_k\| \leq \beta$, *és hogy az* f *célfüggvény Lipschitz folytonosan deriválható, és alulról korlátos a* $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$ *nivóhalmazon. Tegyük fel, hogy a feltételes minimalizálási részfeladat megoldásának* p_k *approximációja eleget tesz az*

$$m_k(0) - m_k(p_k) \geq c_1 \|\nabla f_k\| \min\{\triangle_k, \frac{\|\nabla f_k\|}{\|B_k\|}\} \quad \text{és} \quad \|p_k\| \leq \gamma \triangle_k$$

egyenlőtlenségeknek, ahol $c_1 \in (0, 1]$ *és* $\gamma \geq 1$. *Akkor* $\lim_{k \to \infty} \nabla f_k = 0$, *vagyis az eljárás globálisan konvergens.*

Feltételes optimalizálási feladat

$$\min_{x \in \Omega} f(x), \quad \text{ahol} \quad \Omega = \{x : g_i(x) = 0, i \in \mathcal{E}, g_i(x) \geq 0, i \in \mathcal{I}\},$$

ahol

- $f : \mathbb{R}^n \to \mathbb{R}$ a célfüggvény;
- $\mathcal{E}$ az egyenlőség alakú megszorítások indexének halmaza;
- $\mathcal{I}$ az egyenlőtlenség alakú megszorítások indexének halmaza;

Mi a megoldása a feltételes optimalizálási feladatnak?

Definíció

Az $x^* \in \mathbb{R}^n$ pont a feltételes optimalizáció lokális minimuma ha létezik az x^* olyan $\mathcal{N}$ környezete, ahol $f(x^*) \leq f(x)$ minden $x \in \mathcal{N} \cap \Omega$.

Mi a megoldása a feltételes optimalizálási feladatnak?

Definíció

Az $x^* \in \mathbb{R}^n$ pont a feltételes optimalizáció lokális minimuma ha létezik az x^* olyan $\mathcal{N}$ környezete, ahol $f(x^*) \leq f(x)$ minden $x \in \mathcal{N} \cap \Omega$.

Definíció

Az $x^* \in \mathbb{R}^n$ pont a feltételes optimalizáció szigorú lokális minimuma ha létezik az x^* olyan $\mathcal{N}$ környezete, ahol $f(x^*) < f(x)$ minden $x \in \mathcal{N} \cap \Omega$.

A feltételes optimalizáció klaszifikációja

Konvex programozás: A célfüggvény és a megszorítások is konvexek. Minden lokális optimum globális is egyben.

Lineáris programozás: Mind a célfüggvény, mind a feltételek lineárisak. Tekinthető a konvex optimalizálás részének.

Kvadratikus programozás: A célfüggvény kvadratikus, a feltételek lineárisak.

Nemlineáris programozás: Nemlineáris célfüggvény, feltételek. Rövidítés: *NLO* - Nemnlineáris optimalizálási feladat.

Többcélú optimalizálás: A probléma nem csak egy, hanem több célfüggvénnyel rendelkezik, amelyek nem vonhatóak össze. Nem egyértelmű, hogy mit nevezünk megoldásnak.

Optimalitási feltételek

A feltételes optimalizálási feladat *Lagrange-függvényének* nevezzük az

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i g_i(x).$$

Jelölje az $\mathcal{A}(x)$ indexhalmaz az aktív egyenlőtlenség feltételeket, vagyis

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} : g_i(x) = 0\}.$$

Definíció

Tegyük fel, hogy a feltételes optimalizálási feladat célfüggvénye és feltétel függvényei folytonosan differenciálhatók. A LICQ (linearly independent constraint qualification) regularitási feltétel teljesül az x^* pontban, ha az aktív egyenlőtlenség feltételeinek gradiensei $\{\nabla g_i(x^*) : i \in \mathcal{A}(x^*)\}$ lineárisan függetlenek.

Elsőrendű optimalitási feltétel

Tétel (Karush-Kuhn-Tucker tétele)

Legyen x^* az NLO lokális minimuma és tegyük fel, hogy az x^* pontban teljesül a LICQ regularitási feltétel. Akkor létezik a λ^* Lagrange szorzók vektora úgy, hogy a (x^*, λ^*) rendezett párra érvényes:

$$\begin{aligned}
\nabla_x \mathcal{L}(x^*, \lambda^*) &= 0 \\
g_i(x^*) &= 0, \quad \textit{minden } \ i \in \mathcal{E} \textit{ indexre}, \\
g_i(x^*) &\geq 0, \quad \textit{minden } \ i \in \mathcal{I} \textit{ indexre}, \\
\lambda_i^* &\geq 0, \quad \textit{minden } \ i \in \mathcal{I} \textit{ indexre}, \\
\lambda_i^* g_i(x^*) &= 0, \quad \textit{minden } \ i \in \mathcal{E} \cup \mathcal{I} \textit{ indexre}.
\end{aligned}$$

A fenti feltételeket *KKT feltételeknek* nevezük, még az (x^*, λ^*) pontot *KKT* pontnak nevezzük. A $\lambda_i^* g_i(x^*) = 0$ feltételt a *komplementaritási feltételnek* nevezzük. A $g_i(x^*) = 0$ és $g_i(x^*) \geq 0$ feltételek biztosítják, hogy x^* megengedett legyen.

Elsőrendű optimalitási feltétel

A komplementaritási feltételnek köszönhetően a $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$ felírhatjuk a következő módon:

$$0 = \nabla_x \mathcal{L}(x^*, \lambda^*) = \nabla f(x^*) - \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla g_i(x^*).$$

Az *NLO* egy adott x^* lokális minimumához több λ^* vektor tartozhat úgy, hogy az (x^*, λ^*) pont teljesíti a *KKT* feltételeket. Viszont, ha teljesül a *LICQ* regularitási feltétel, akkor a λ^* egységes.

Másodrendű optimalitási feltételek

Tételezzük fel, hogy az (x^*, λ^*) eleget tesz a KKT feltételeknek.
Definiáljuk a következő halmazt:

$$w \in F(\lambda^*) = \begin{cases} \nabla g_i(x^*)^T w = 0 & \text{minden } i \in \mathcal{E} \text{ indexre} \\ \nabla g_i(x^*)^T w = 0 & \text{minden } i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ melyre } \lambda_i^* > 0 \\ \nabla g_i(x^*)^T w \geq 0 & \text{minden } i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ melyre } \lambda_i^* = 0 \end{cases}$$

Másodrendű optimalitási feltételek

Tételezzük fel, hogy az (x^*, λ^*) eleget tesz a *KKT* feltételeknek. Definiáljuk a következő halmazt:

$$w \in F(\lambda^*) = \begin{cases} \nabla g_i(x^*)^T w = 0 & \text{minden } i \in \mathcal{E} \text{ indexre} \\ \nabla g_i(x^*)^T w = 0 & \text{minden } i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ melyre } \lambda_i^* > 0 \\ \nabla g_i(x^*)^T w \geq 0 & \text{minden } i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ melyre } \lambda_i^* = 0 \end{cases}$$

Tétel (Másodrendű szükséges feltétel)

Tegyük fel, hogy x^* az NLO lokális minimuma és a LICQ regularitási feltételek teljesülnek. Legyen λ^* a Lagrange szorzók vektora úgy, hogy az (x^*, λ^*) eleget tegyen a KKT feltételeknek. Akkor

$$w^T H(x^*, \lambda^*) w \geq 0 \quad \textit{minden} \quad w \in F(\lambda^*),$$

ahol $H(x^*, \lambda^*)$ a Lagrange-függvény Hesse mátrixa az (x^*, λ^*) pontban.

Másodrendű optimalitási feltételek

Tegyük fel, hogy egy $x^* \in \mathbb{R}^n$ megengedett pont mellett létezik a λ^* Lagrange szorzók vektora, melyekben a a KKT feltételek teljesülnek. Szintén tételezzük fel, hogy minden $w \in F(\lambda^*), w \neq 0$ vektorra teljesül

$$w^T H(x^*, \lambda^*) w > 0.$$

Akkor az x^* pont az NLO szigorú lokális minimuma.

Konvex programozás (KO)

Feltételes optimializálási feladat, amelyben az $f : \mathbb{R}^n \to \mathbb{R}$ célfüggvény konvex és az Ω megszorítások halmaza is konvex. A *KO*-ra érvényes:

- a KO lokális optimuma globális optimum;
- A *KKT* feltétel elegendő az optimalitáshoz;

Lineáris programozás

A lineáris programozás feladata (LP): lineáris korlátozó feltételek mellett keresendő egy lineáris függvény *célfüggvény* szélsőértéke (minimuma vagy maximuma).

Lineáris programozás

A lineáris programozás feladata (LP): lineáris korlátozó feltételek mellett keresendő egy lineáris függvény *célfüggvény* szélsőértéke (minimuma vagy maximuma).

Példa (Termékösszetétel optimalizálás)

Tekintsünk egy gyárat, amely n-féle terméket gyárt. Tegyük fel, hogy a termékek gyártásához m-féle erőforrás szükséges. Legyenek ismertek a gyártási folyamat következő jellemzői:

- a_{ij} *az i-edik erőforrásból a j-edik termék egységének az előállításához szükséges mennyiség;*
- b_i *az i-edik erőforrásból az optimalizálási időszakasz alatt rendelkezésre álló mennyiség,*
- c_j *a j-edik termék egységének a gyártási haszna.*

Lineáris programozás

A termékösszetétel optimalizálása ekkor abból áll, hogy az egyes gyártható termékekből annyit akarunk gyártani, amennyit a rendelkezésre álló erőforrások még lehetővé tesznek, miközben összességében a lehető legnagyobb gyártási hasznot kívánjuk elérni. Az így megfogalmazott probléma matematikai modellezéséhez be kell még vezetni az x_j döntési változókat:

x_j a j-edik termékből az optimalizálási időszakasz alatt gyártandó mennyiség.

Az így megfogalmazott problémát matematikailag az alábbi LP feladattal lehet leírni:

Lineáris programozás

$$
\begin{array}{lllll}
c_1x_1 & +c_2x_2 & +\cdots & +c_nx_n & \rightarrow max \\
a_{11}x_1 & +a_{12}x_2 & +\cdots & +a_{1n}x_n & \leq b_1 \\
a_{21}x_1 & +a_{22}x_2 & +\cdots & +a_{2n}x_n & \leq b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{m1}x_1 & +a_{m2}x_2 & +\cdots & +a_{mn}x_n & \leq b_m \\
x_1 \geq 0 & ,x_2 \geq 0 & ,\cdots & ,x_n \geq 0 &
\end{array}
$$

Lineáris programozás

$$
\begin{array}{lllll}
c_1x_1 & +c_2x_2 & +\cdots & +c_nx_n & \rightarrow max \\
a_{11}x_1 & +a_{12}x_2 & +\cdots & +a_{1n}x_n & \leq b_1 \\
a_{21}x_1 & +a_{22}x_2 & +\cdots & +a_{2n}x_n & \leq b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{m1}x_1 & +a_{m2}x_2 & +\cdots & +a_{mn}x_n & \leq b_m \\
x_1 \geq 0 & ,x_2 \geq 0 & ,\ldots & ,x_n \geq 0 &
\end{array}
$$

Az előző matematikai modellt mátrixos alakban is fel tudjuk írni:

$$\max c^T x, \text{ f.h. } Ax \leq b, \ x \geq 0.$$

Lineáris programozás

További példák

- takarmányösszetétel optimalizálás;
- szállítási feladat;
- repülőgépforgalom megszervezése;
- rendőrségi őrjáratok megszervezése;
- szemételhordás megszervezése;
- szennyeződés kontrolja;
- befektetések tervezése;

Szimplex módszer

Az *LP* standard alakja

A lineáris programozási feladat standard alakú, ha minden lineáris korlátozó feltétele egyenlőség alakú, minden változójára elő van írva a nemnegativitási korlátozás és a célfüggvénye maximalizálandó:

$$\max c^T x, \text{ f.h. } Ax = b, \; x \geq 0.$$

Szimplex módszer

Az *LP* standard alakja

A lineáris programozási feladat standard alakú, ha minden lineáris korlátozó feltétele egyenlőség alakú, minden változójára elő van írva a nemnegativitási korlátozás és a célfüggvénye maximalizálandó:

$$\max c^T x, \text{ f.h. } Ax = b, \; x \geq 0.$$

Minden lineáris programozási feladat ekvivalens átalakításokkal standard alakra transzformálható.

Szimplex módszer

Minden lineáris programozási feladat ekvivalens átalakításokkal standard alakra transzformálható.

- Kisebb vagy egyenlő alakú korlátozó feltétel egyenlőséggé alakítása: nemnegatív segédváltozó bevezetésével.
- Nagyobb vagy egyenlő alakú korlátozó feltétel egyenlőséggé alakítása: a nagyobb vagy egyenlő alakú korlátozó feltételhez vezetünk be egy nemnegatív segédváltozót, melyet ki kell vonni a feltétel bal oldalából.
- Szabad változó helyettesítése nemnegatív változókkal: ha egy változó, mondjuk x nincs nemnegativitással korlátozva (szabad változó), akkor előállítható mint $x = x^+ - x^-$ alakban, ahol $x^+ = \max x, 0$, $x^- = \max -x, 0$.

Szimplex módszer

- Minimalizálandó célfüggvény helyettesítése maximalizálandó célfüggvénnyel: ha egy célfüggvényt minimalizálni kell, az ekvivalens azzal, mint hogyha a célfüggvény mínusz egyszeresét kell maximalizálni.

Szimplex módszer

- Minimalizálandó célfüggvény helyettesítése maximalizálandó célfüggvénnyel: ha egy célfüggvényt minimalizálni kell, az ekvivalens azzal, mint hogyha a célfüggvény mínusz egyszeresét kell maximalizálni.

Az *LP* megoldhatósága

- Nincs megoldása a feladatnak: a megszorítások által definiált konvex poliéder üres halmaz.
- Nincs véges optimuma a feladatnak: a megszorítások által definiált konvex poliéder nem korlátos és a célfüggvény növekedési iránya egybeesik a konvex poliéder "nemkorlátossági irányával".

Szimplex módszer

Az *LP* megoldhatósága

- A feladat véges optimuma a konvex poliéder egy vagy több csúcspontján valósul meg: ha a *LP*-nek létezik megoldása és véges optimuma, akkor a véges optimum mindig megvalósul a korlátozó feltételek által létrehozott konvex poliéder legalább egy csúcsán is.

Szimplex módszer

Az *LP* megoldhatósága

- A feladat véges optimuma a konvex poliéder egy vagy több csúcspontján valósul meg: ha a *LP*-nek létezik megoldása és véges optimuma, akkor a véges optimum mindig megvalósul a korlátozó feltételek által létrehozott konvex poliéder legalább egy csúcsán is.

A szimplex módszer egy olyan eljárás, amelynek bemenetele az *LP* egy megengedett megoldása (induló megoldás) és amely mindegyik lépésben egy olyan megengedett megoldást talál meg, amely közelebb van az *LP* optimális megoldásához. Ha az *LP* problémának létezik megoldása, a szimplex módszer véges számú lépésben találja meg az optimális megoldást. A szimplex módszer mindegyik lépésben a megengedett megoldások halmazának (konvex poliéder) egyik csúcsát (bázis megoldást) találja meg.

Dualitás az *LP*-ben

A

$$\begin{aligned} c^T x \rightarrow & \quad \min & \qquad b^T y \rightarrow & \quad \max \\ Ax = & \quad b & \qquad A^T y \leq & \quad c \\ x \geq & \quad 0 \end{aligned}$$

LP feladatok egymásnak a *duáljai*. A kiinduló feladatot *primál*, a belőle származtatott feladatot *duál* feladatnak nevezzük.

Tétel

A duál feladat duálja maga a kiinduló primál feladat.

Tétel (Dualitás tétel)

Ha egy primál-duál feladat pár egyikének létezik megengedett megoldása és véges optimuma, akkor ugyanez fennáll a másikra is, és a két feladat optimum értéke egyenlő.

Dualitás az *LP*-ben

A primál és duál *KKT* feltételei

$$\begin{aligned} A^T\lambda + s &= c \\ Ax &= b \\ x &\geq 0 \\ s &\geq 0 \\ x_i s_i &= 0, \end{aligned} \qquad \begin{aligned} Ax &= b \\ A^T\lambda &\leq c \\ x &\geq 0 \\ x_i(c - A^T\pi)_i &= 0 \end{aligned}$$

$i = 1, 2, \ldots, n$. Ha $s = c - A^T\lambda$ a duál *KKT* feltételekben, megkapjuk a primál *KKT* feltételeit.

Belsőpontos módszerek

A primál és a duál KKT feltételeit felírhatjuk a következő módon:

$$F(x,\lambda,s) = \begin{bmatrix} A^T\lambda + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0,$$
$$(x,s) \geq 0,$$

ahol $X = diag(x_1, x_2, \ldots, x_n)$ és $S = diag(s_1, s_2, \ldots, s_n)$. A *belsőpontos* módszerek, olyan (x^k, λ^k, s^k) iterációs sorozatot hoznak létre, amelyekre érvényes $x^k > 0, s^k > 0$. Mindegyik iterációban a fenti nemlineáris egyenletrendszert kell Newton-módszerrel megoldani úgy, hogy $x^k > 0, s^k > 0$.

Belsőpontos módszerek

Newton-lépés

A következő iterációt a következő módon számoljuk ki

$$(x, \lambda, s) + \alpha(\triangle x, \triangle \lambda, \triangle s),$$

ahol (x, λ, s) a pillanatnyi iteráció, $\alpha \in (0, 1]$ a lépéshossz, $(\triangle x, \triangle \lambda, \triangle s)$ a kutatási irány amit a következő egyenletrendszer megoldásával határozunk meg:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \triangle x \\ \triangle \lambda \\ \triangle s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe \end{bmatrix}$$

Az $\alpha = 1$ teljes lépéshossz sűrűn nem elfogadható, mivel akkor az $(x, s) \geq 0$ feltétel nem teljesül. Nagyon sűrűn $\alpha \ll 1$, hogy teljesüljön $(x, s) > 0$, ezért módosítjuk a Newton-módszert.

Belsőpontos módszerek

Módosított Newton-lépés

Legyen $\sigma \in [0,1]$ a *központosító paraméter*, még a μ dualitás mértékét definiáljuk a

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i s_i = \frac{x^T s}{n}.$$

Ezek után mindegyik lépésben a következő módosított Newton-rendszert kell megoldani:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \triangle x \\ \triangle \lambda \\ \triangle s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe + \sigma\mu e \end{bmatrix}.$$

Belsőpontos módszer I

Bemenetel: Kezdő iteráció $(x_0\lambda_0, s_0)$ belsőpont, mely eleme a $\{(x\lambda, s) : Ax = b, A^T\lambda + s = c, (x, s) > 0\}$ halmaznak;

1: $k = 0$;

2: **while** a megállási feltétel nem teljesül

3: Kutatási irány meghatározása: old meg a

4:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S_k & 0 & X_k \end{bmatrix} \begin{bmatrix} \triangle x_k \\ \triangle \lambda_k \\ \triangle s_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X_k S_k e + \sigma_k \mu_k e \end{bmatrix}.$$

egyenletrendszert, ahol $\sigma \in [0, 1]$ és $\mu = \frac{x^T s}{n}$.

5: Lépéshossz meghatározása: határozd meg az $\alpha_k \in (0, 1]$ úgy, hogy $(x_{k+1}, s_{k+1}) > 0$, ahol:

Belsőpontos módszer II

6: Következő iteráció:

$$(x_{k+1}, \lambda_{k+1}, s_{k+1}) = (x_k, \lambda_k, s_k) + \alpha_k(\triangle x_k, \triangle \lambda_k, \triangle s_k)$$

7: $k = k + 1;$

8: **end while**

Szimplex és belsőpontos módszer a Matlab-ban

A linprog függvény segítségével tudjuk alkalmazni a szimplex és belsőpontos módszert az LP feladatokra.

Szimplex és belsőpontos módszer a Matlab-ban

A linprog függvény segítségével tudjuk alkalmazni a szimplex és belsőpontos módszert az *LP* feladatokra.

8. Feladat

Matlab segítségével oldja meg a következő LP feladatot:

$$
\begin{array}{rrrrrr}
80x_1+ & 60x_2+ & 20x_3+ & 100x_4+ & 20x_5 \rightarrow & max \\
x_1+ & 4x_2+ & x_3+ & 2x_4 & \leq & 90 \\
 & 3x_2 & + & x_4+ & 2x_5 \leq & 50 \\
x_1+ & 2x_2 & + & x_4+ & x_5 = & 60 \\
x_1 & + & x_3 & + & x_5 = & 80 \\
x_1 \geq & 40, & x_2 \geq & 0, & x_3 \geq & 20 \\
 & & x_4 \geq & 2, & x_5 \geq & 0
\end{array}
$$

Nemlineáris programozás (*NLO*) alapjai

$$\min_{x \in \Omega} f(x), \quad \text{ahol} \quad \Omega = \{x : g_i(x) = 0, i \in \mathcal{E}, g_i(x) \geq 0, i \in \mathcal{I}\},$$

ahol

- $f : \mathbb{R}^n \to \mathbb{R}$ deriválható függvény (célfüggvény);
- $g_i : \mathbb{R}^n \to \mathbb{R}, i \in \mathcal{E} \cup \mathcal{I}$ deriválható, függvények (feltételek, megszorítások)
- $\mathcal{E}$ az egyenlőség alakú megszorítások indexének halmaza;
- $\mathcal{I}$ az egyenlőtlenség alakú megszorítások indexének halmaza;
- a célfüggvény vagy legalább egy feltételfüggvény nemlineáris.

NLO feladat megoldására szolgáló módszerek

- büntető függvény módszerek;
- kibővített Lagrange-függvény módszerek;
- belsőpontos módszerek;
- szekvenciális kvadratikus programozás

Szkevenciális kvadratikus programozás *SQP*

Egy általánosan használható algoritmus nemlineáris optimalizálási problémák megoldására a következő feltételekkel:

- a feladat nem túl nagy;
- a függvények és gradienseik meghatározhatók megfelelően nagy pontossággal;

Az *SQP* egy $\{x_k\}$ iterációs sorozatot generál kvadratikus programozási részfeladat megoldásáva. Használható egyaránt a vonalmenti minimalizálás és a trust-region kereten belül.

Szekvenciális kvadratikus programozás

A $\min_{x\in\Omega} f(x)$, ahol $\Omega = \{x : g_i(x) = 0, i \in \mathcal{E}, g_i(x) \geq 0, i \in \mathcal{I}\}$
NLO-hoz hozzárendelünk egy kvadratikus programozási részfeladatot:

$$\begin{aligned} \min_p \quad & f_k + \nabla f_k^T p + \frac{1}{2} p^T H(x) p \\ \text{feltéve, hogy} \quad & \nabla g_i(x_k) p + g_i(x_k) = 0, \ i \in \mathcal{E}, \\ & \nabla g_i(x_k) p + g_i(x_k) \geq 0, \ i \in \mathcal{I}. \end{aligned}$$

Ezt a kvadratikus programozási részfeladatot az erre szolgáló technikák segítségével tudjuk megoldani.
Az új iteráció

$$(x_k + p_k, \lambda_{k+1}),$$

ahol p_k a kvadratikus programozási részfeladat megoldása és λ_{k+1} a megfelelő Lagrange szorzó.

SQP a Matlab-ban

A fmincon függvény az SQP módszert alkalmazza az *NLO* feladatokra.

9. Feladat

Matlab segítségével oldjuk meg az SQP módszerrel $(3, 1)$ *kezdőponttal a következő NLO feladatot.*

$$
\begin{aligned}
\min \quad & x_1^2 + x_2^2 \\
f.h. \quad & x_1 + x_2 - 1 \geq 0 \\
& x_1^2 + x_2^2 - 1 \geq 0 \\
& 9x_1^2 + x_2^2 - 9 \geq 0 \\
& x_1^2 - x_2 \geq 0 \\
& x_2^2 - x_1 \geq 0
\end{aligned}
$$