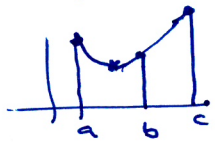
Minimalizace (hledání minima, ale i maxima) funkce

- speciální algoritmy pro funkce jedné proměnné
  vs. univerzální metody pro fce lib. počtu proměnných
- problém nalezení <u>globálního minima</u> (pro maximum $-f$)
  namísto lokálního - nutno vhodně „nastřelit",
  ovšem pro fce mnoha proměnných velmi obtížné
- efektivní algoritmy potřebují kromě $f(x)$ také $\frac{\partial f}{\partial x_i}$
  což může být mnohdy problém - např. pokud $f(x)$ je počítáno složitým algoritmem a navíc pokud je chyba „náhodná", jako např. v MC metodách

jednorozměrné funkce:

a) nemáme-li k dispozici derivaci lze použít tzv. uzávorkování (bracketing) minima:



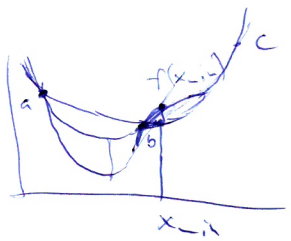
pro $a<b<c$ bude $f(a)>f(b)<f(c)$
$\Rightarrow$ pak nutně leží minimum někde v $\langle a,c\rangle$
pokud není singulární

pokud má funkce spojité první a druhé derivace $\rightarrow$ Brentova metoda
jinak použij obdobu půlení intervalu, ovšem založené na zlatém řezu

- <u>Brentova metoda</u> založena na nalezení minima parabolí jdoucí $(a,f(a))$, $(b,f(b))$ a $(c,f(c))$

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

a vhodným výběrem dalších bodů



b) jsou-li k dispozici derivace
lze je využít k optimálnějšímu hledání minima (viz Numerical Recipes)

Poznámka k přesnosti: Je-li v b minimum, pak
z $f(x) \approx f(b) + \frac{1}{2} f''(b)(x-b)^2$ plyne, že je-li $\frac{|f(x)-f(b)|}{|f(b)|} \sim \varepsilon$
nelze obecně očekávat, že $|x-b|$ bude lepší než $\sqrt{\varepsilon}$
pro $|x-b| < \sqrt{\varepsilon}\sqrt{\frac{2|f(b)|}{f''(b)}}$ bude $f(x) \approx f(b)$ neboť $|x-b|^2 \sim \varepsilon$

funkce více proměnných – není obdoba uzávorkování!

a) **sestupná metoda** založená na porovnání hodnot fce ve vrcholech simplexu – nepotřebuje derivace

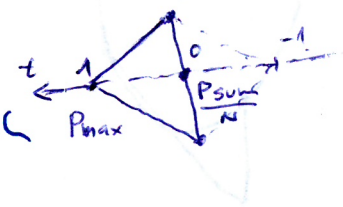
- jednoduchá, poměrně robustní metoda
- simplex v N dimenzích má N+1 vrcholů a nenulový objem (ve 2D jde o trojúhelník, 3D čtyřstěn atd.)
- kromě počátečního bodu $P_0 = (x_0^1, \dots, x_0^N)$ je nutno zadat dalších N bodů definujících počáteční simplex typicky pomocí počátečních posunutí v jednotlivých prom.

tj.
$$P_i = P_0 + \lambda_i e_i \quad , \quad i = 1, \dots, N$$

kde e_i je jednotkový vektor ve směru x^i
a λ_i je počáteční posunutí v tomto směru
(λ_i nemusí být ve všech směrech stejné, záleží na problému)

- základní strategie: „zrcadli" bod, kde je f(p) největší pomocí ostatních bodů, případně kontrahuj

„zrcadlení" není zrcadlení v pravém slova smyslu, nový bod bude na spojnici „těžiště" ostatních bodů a bodu, kde je maximum



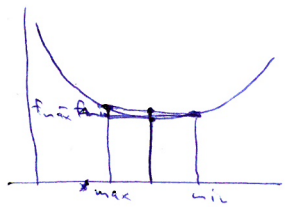
$$P_{new} = \frac{P_{sum}}{N}(1-t) + P_{max}\, t$$

pokud $f(P_{new}) < f(P_{max})$ pak P_{max} nahraď tímto P_{new}

kde $P_{sum} = \sum_{\substack{i=0 \\ i \neq max}}^{N} P_i$

Amoeba z numerických receptů (hledání minima)

1) **test přesnosti**: algoritmus skončí, pokud



$$\frac{|f_{max} - f_{min}|}{\frac{|f_{max}| + |f_{min}| + \varepsilon}{2}} < tol,$$

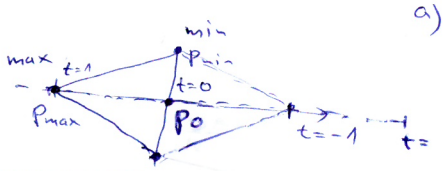
kde tol je předepsaná tolerance (typicky $10^{-3} - 10^{-8}$ podle potřeby) a f_{max} a f_{min} jsou maximum a minimum v bodech simplexu. $\varepsilon \sim 10^{-10}$ je malé kladné číslo zaručující dělení nulou, pokud

$$f_{max} = f_{min} = 0$$

! problém, pokud se při výběru bodů náhodou trefíme tam, kde má fce stejné hodnoty ⇒ automaticky konec

⇒ je potřeba zkusit více počátečních bodů a kroků

2) volba nového bodu



a) nejprve zkusíme bod středově souměrný vůči p_0

$$p_{try} = p_0 + p_0 - p_{max} = 2p_0 - p_{max}$$

(volba $t = -1$)

obecně

$$p_{try}(t) = p_0(1-t) + p_{max}t \quad , \text{kde} \quad p_0 = \frac{1}{N}\sum_{\substack{i=0 \\ i \neq imax}}^{N} p_i$$

pozor v algoritmu se používá $p_{sum} = \sum_{i=0}^{N} p_i$ (tj. včetně p_{imax})

a tedy $$p_{try} = p_{sum}\frac{1-t}{N} + \left(t - \frac{1-t}{N}\right)p_{max}$$

- postupně se zkouší $t = -1$, je-li tam minimum, tak $t = -2$ (v Amoebě se volá AmoebaTry s $t=2$, protože už jsme se posunuli)

pokud tam není minimum, vrátíme do $t = -\frac{1}{2}$, pokud je hodnota velká (větší než druhé maximum), a pokud i tam je hodnota velká, zmenší se celý simplex směrem k minimu