

Numerické metody

Doc. RNDr. Libor Čermák, CSc. RNDr. Rudolf Hlavička, CSc.

Ústav matematiky
Fakulta strojního inženýrství
Vysoké učení technické v Brně

23. ledna 2006

Obsah

- 6 Optimalizace
 - Úvod
 - Jednorozměrná minimalizace
 - Minimalizace funkce více proměnných
 - Literatura

Optimalizační úlohy se zabývají výběrem „nejlepších řešení“ z dané množiny „možných řešení“. Matematicky můžeme optimalizační úlohu formulovat jako nalezení prvku $\mathbf{x}^* \in M$ takového, že pro libovolný prvek $\mathbf{x} \in M$ platí

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in M, \quad (6.1)$$

kde $f : M \mapsto \mathbb{R}$ je minimalizovaná (někdy se také říká **účelová** nebo **cílová** nebo **kriteriální**) **funkce** a M je **množina přípustných řešení**. Jestliže přípustným řešením může být každý bod $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ n -rozměrného Euklidova prostoru \mathbb{R}^n , tj. $M = \mathbb{R}^n$, hovoříme o **nepodmíněné optimalizaci**. O funkci f budeme předpokládat, že je spojitá (případně i se svými prvními a dalšími derivacemi).

Optimalizační úloha (6.1) se nazývá **úlohou globální optimalizace**. My se v této kapitole omezíme na jednodušší **úlohu lokální optimalizace** spočívající v nalezení **lokálního minima**, tj. prvku $\mathbf{x}^* \in M$ takového, že pro libovolný prvek $\mathbf{x} \in M$ platí

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in M \cap O(\mathbf{x}^*), \quad (6.2)$$

kde $O(\mathbf{x}^*)$ je nějaké okolí bodu \mathbf{x}^* .

Podrobněji si všimneme dvou speciálních úloh: v odstavci 5 se seznámíme s metodami pro minimalizaci funkce jedné proměnné na intervalu $\langle a, b \rangle$ a v odstavci 20 se budeme věnovat metodám nepodmíněné minimalizace funkce více proměnných.

Poznámka. Určení maxima funkce $g(\mathbf{x})$ můžeme převést na úlohu určení minima funkce $f(\mathbf{x}) = -g(\mathbf{x})$. \square

Obsah

- 6 Optimalizace
 - Úvod
 - Jednorozměrná minimalizace
 - Minimalizace funkce více proměnných
 - Literatura

V tomto odstavci uvedeme metody pro přibližné určení bodu x^* lokálního minima funkce $f(x)$ na intervalu $\langle a, b \rangle$. Jestliže je funkce f na intervalu $\langle a, b \rangle$ **unimodální**, tj. když má v intervalu $\langle a, b \rangle$ jediné minimum, pak ho (dále uvedenými metodami přibližně) najdeme. Pokud však má funkce f na intervalu $\langle a, b \rangle$ více lokálních minim, najdeme jedno z nich (nemusí jít o minimum globální).

Intervalovou bisekci použít nemůžeme: i když známe $f(a)$, $f(b)$ a $f((a+b)/2)$, nedokážeme rozhodnout, ve které polovině intervalu $\langle a, b \rangle$ minimum leží.

Použít lze intervalovou trisekci. Nechť $h = (b-a)/3$, takže $u = a + h$ a $v = b - h$ dělí interval na tři stejné části. Předpokládejme, že $f(u) < f(v)$. Pak minimum jistě leží vlevo od v , takže b nahradíme pomocí v . Tím se délka intervalu (obsahujícího minimum) zkrátí na dvě třetiny své původní délky. Bod u se však stane středem nového intervalu a nebude proto v dalším kroku využitelný. Funkci f tedy musíme vyhodnocovat v každém kroku dvakrát. To je neefektivní.

Metoda zlatého řezu je založena na šikovnějším výběru dělicích bodů u a v . Necht' $h = \varrho(b - a)$, kde ϱ je číslo o něco větší než $1/3$, jehož přesnou hodnotu teprve určíme. Pak body $u = a + h$ a $v = b - h$ dělí interval $\langle a, b \rangle$ na tři nestejně části. V prvním kroku vyhodnotíme $f(u)$ a $f(v)$. Předpokládejme, že $f(u) < f(v)$. Pak víme, že minimum je mezi a a v . Nahradíme b pomocí v a proces opakujeme. Když zvolíme „správnou“ hodnotu ϱ , bod u bude ve „správné pozici“ použitelné v příštím kroku. Po prvním kroku se tak funkce f bude vyhodnocovat v každém kroku už jen jednou.

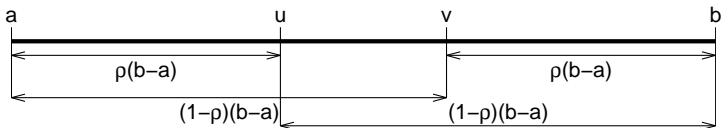
Jak tedy zvolit ϱ ? Tak, aby bod u hrál v redukovaném intervalu $\langle a, v \rangle$ stejnou roli jako bod v v původním intervalu $\langle a, b \rangle$, tj. aby poměr délky intervalu $\langle a, u \rangle$ k délce intervalu $\langle a, v \rangle$ byl stejný jako poměr délky intervalu $\langle a, v \rangle$ k délce intervalu $\langle a, b \rangle$,

$$\frac{u - a}{v - a} = \frac{v - a}{b - a} \quad \Longleftrightarrow \quad \frac{\varrho}{1 - \varrho} = \frac{1 - \varrho}{1} \quad \Longleftrightarrow \quad \varrho^2 - 3\varrho + 1 = 0.$$

Vyhovující řešení je

$$\varrho = (3 - \sqrt{5})/2 \approx 0,382, \quad \text{kde} \quad \tau = 1 - \varrho = (\sqrt{5} - 1)/2 \approx 0,618$$

je číslo známé jako **poměr zlatého řezu**.



Obr. 6.1: Zlatý řez

Poznámka (**O zlatém řezu**). Říkáme, že bod dělí interval v poměru zlatého řezu, když dva nově vzniklé subintervaly mají tuto vlastnost: poměr délky kratšího subintervalu k délce delšího subintervalu je stejný jako poměr délky delšího subintervalu k délce celého intervalu. Z výše uvedené konstrukce je zřejmé, že bod u (ale také bod v) dělí interval $\langle a, b \rangle$ v poměru zlatého řezu.

Přimeňme si, že s číslem τ jsme se setkali již v kapitole *řešení nelineárních rovnic*, kde jsme uvedli, že rychlost konvergence metody sečen $p = 1 + \tau \doteq 1,618$. \square

Zatím jsme předpokládali, že $f(u) < f(v)$. V opačném případě, tj. když $f(u) \geq f(v)$, leží minimum v intervalu $\langle u, b \rangle$, takže a nahradíme pomocí v . Snadno ověříme, že v redukovaném intervalu $\langle u, b \rangle$ bude mít v stejnou roli jako mělo u v původním intervalu $\langle a, b \rangle$, takže hodnotu funkce f na redukovaném intervalu opět stačí počítat jen jednou.

Délka redukovaného intervalu je τ -krát menší než délka původního intervalu. Z výchozího intervalu $\langle a_0, b_0 \rangle = \langle a, b \rangle$ tak postupně sestrojíme intervaly $\langle a_1, b_1 \rangle \supset \langle a_2, b_2 \rangle \supset \dots$, které obsahují minimum a jejichž délka je v každém kroku redukována faktorem τ . Na výchozím intervalu $\langle a_0, b_0 \rangle$ určíme $u_0 = a + \varrho(b - a)$, $v_0 = b - \varrho(b - a)$ a vypočteme $f(u_0)$, $f(v_0)$. Interval $\langle a_{k+1}, b_{k+1} \rangle$, $k = 0, 1, \dots$, dostaneme pomocí a_k , b_k , u_k , v_k a již dříve vypočtených hodnot $f(u_k)$, $f(v_k)$ takto:

1) když $f(u_k) < f(v_k)$, pak

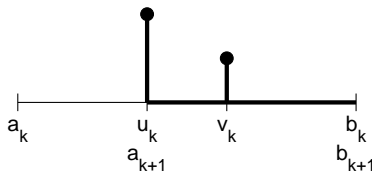
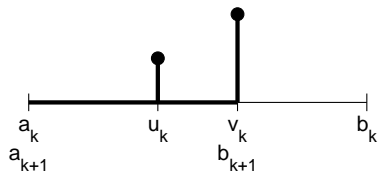
$$a_{k+1} := a_k, \quad b_{k+1} := v_k, \quad v_{k+1} := u_k, \quad u_{k+1} := a_{k+1} + \varrho(b_{k+1} - a_{k+1})$$

a vypočteme $f(u_{k+1})$;

2) v opačném případě, tj. když $f(u_k) \geq f(v_k)$, provedeme

$$a_{k+1} := u_k, \quad b_{k+1} := b_k, \quad u_{k+1} := v_k, \quad v_{k+1} := b_{k+1} - \varrho(b_{k+1} - a_{k+1})$$

a vypočteme $f(v_{k+1})$.



Obr. 6.2: Metoda zlatého řezu

Po k krocích leží minimum v intervalu $I_k := \langle a_k, b_k \rangle$ délky

$$|I_k| = b_k - a_k = \tau(b_{k-1} - a_{k-1}) = \dots = \tau^k(b_0 - a_0).$$

Střed x_{k+1} intervalu $\langle a_k, b_k \rangle$ aproximuje minimum x^* s chybou

$$|x_{k+1} - x^*| \leq \frac{1}{2}(b_k - a_k) = \frac{1}{2}\tau^k(b_0 - a_0). \quad (6.3)$$

Pro $k \rightarrow \infty$ zřejmě $|I_k| \rightarrow 0$ a $x_k \rightarrow x^*$.

Konvergence metody zlatého řezu je poměrně pomalá. Proto je v blízkosti minima účelné přejít na rychleji konvergentní metodu. Jednou z možností je metoda kvadratické interpolace, s níž se také seznámíme. Předtím ale

Příklad 6.1. Metodou zlatého řezu určíme minimum funkce

$f(x) = x^4 - 3x^3 + x + 7$. Jako počáteční zvolíme interval $\langle a_0, b_0 \rangle = \langle 1, 3 \rangle$.

Výpočet provedeme s přesností $\varepsilon = 10^{-3}$, tj. když $b_k - a_k < 2\varepsilon$, položíme $x_{k+1} = (a_k + b_k)/2$. Výpočet je zaznamenán v následující tabulce. Podtržením jsou vyznačeny ty vnitřní body, v nichž se počítá hodnota účelové funkce.

k	a_k	u_k	v_k	b_k	$f(u_k)$		$f(v_k)$
0	1,0000	<u>1,7639</u>	<u>2,2361</u>	3,0000	1,979900	\geq	0,695048
1	1,7639	2,2361	<u>2,5279</u>	3,0000	0,695048	$<$	1,901312
2	1,7639	<u>2,0557</u>	2,2361	2,5279	0,852324	\geq	0,695048
3	2,0557	2,2361	<u>2,3475</u>	2,5279	0,695048	$<$	0,906510
4	2,0557	<u>2,1672</u>	2,2361	2,3475	0,690296	$<$	0,695048
5	2,0557	<u>2,1246</u>	2,1672	2,2361	0,729249	\geq	0,690296
\vdots							
14	2,1973	2,1982	<u>2,1988</u>	2,1997	0,681572	$<$	0,681575
15	2,1973			2,1988			

Požadovaná přesnost je dosažena pro $k = 15$, takže (po zaokrouhlení na 3 desetinné cifry) $x_{16} \doteq 2,198$. Hodnota účelové funkce se počítá celkem 16-krát. Protože přesná hodnota $x^* \doteq 2,198266$, má x_{16} všechny cifry platné. \square

Metoda kvadratické interpolace. Předpokládejme, že minimum leží v intervalu $\langle a_k, b_k \rangle$, a že v nějakém jeho vnitřním bodě c_k hodnota funkce f nepřesáhne hodnoty $f(a_k)$, $f(b_k)$ v krajních bodech a_k , b_k , tj. že

$$\text{pro } a_k < c_k < b_k \text{ platí } f(c_k) \leq \min\{f(a_k); f(b_k)\}. \quad (6.4)$$

Body $[a_k, f(a_k)]$, $[c_k, f(c_k)]$ a $[b_k, f(b_k)]$ proložíme parabolou $P_2(x)$ (kvadratický interpolační polynom) a bod x_{k+1} jejího minima považujeme za další aproximaci x^* . Vzorec pro výpočet x_{k+1} dostaneme řešením lineární rovnice $P_2'(x_{k+1}) = 0$. Dá se ukázat, že

$$x_{k+1} = c_k - \frac{1}{2} \frac{(c_k - a_k)^2[f(c_k) - f(b_k)] - (c_k - b_k)^2[f(c_k) - f(a_k)]}{(c_k - a_k)[f(c_k) - f(b_k)] - (c_k - b_k)[f(c_k) - f(a_k)]}. \quad (6.5)$$

Z (6.4) plyne, že $x_{k+1} \in (a_k, b_k)$. Když náhodou $x_{k+1} = c_k$, vložíme do x_{k+1} jiný vnitřní bod intervalu $\langle a_k, b_k \rangle$. S tímto slabým místem metody kvadratické interpolace (a s dalšími, zde nezmiňnými nedostatky) se úspěšně vyrovnává Brentova metoda. Stručná zmínka o ní je uvedena v následujícím textu.

Z bodů a_k , b_k , c_k a x_{k+1} pak vybereme nový interval (a_{k+1}, b_{k+1}) obsahující minimum a bod c_{k+1} splňující podmínku (6.4), tentokrát pro index $k + 1$.

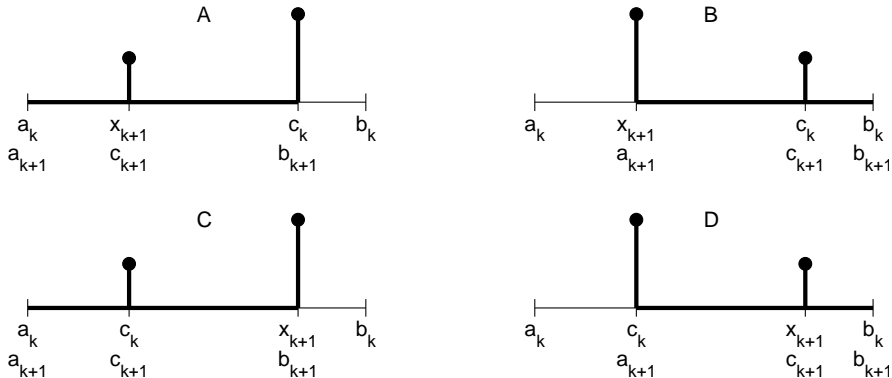
Postupujeme podle následujících pravidel:

$$\text{A: } x_{k+1} < c_k \quad \text{a} \quad f(x_{k+1}) < f(c_k) \quad \implies \quad a_{k+1} = a_k, \quad c_{k+1} = x_{k+1}, \\ b_{k+1} = c_k,$$

$$\text{B: } x_{k+1} < c_k \quad \text{a} \quad f(x_{k+1}) \geq f(c_k) \quad \implies \quad a_{k+1} = x_{k+1}, \quad c_{k+1} = c_k, \\ b_{k+1} = b_k,$$

$$\text{C: } c_k < x_{k+1} \quad \text{a} \quad f(c_k) < f(x_{k+1}) \quad \implies \quad a_{k+1} = a_k, \quad c_{k+1} = c_k, \\ b_{k+1} = x_{k+1},$$

$$\text{D: } c_k < x_{k+1} \quad \text{a} \quad f(c_k) \geq f(x_{k+1}) \quad \implies \quad a_{k+1} = c_k, \quad c_{k+1} = x_{k+1}, \\ b_{k+1} = b_k.$$



Obr. 6.3: Metoda kvadratické interpolace

Výpočet ukončíme a x_{k+1} považujeme za dostatečně dobrou aproximaci minima x^* , když je splněno některé ze stop kriterií

$$\begin{aligned} |x_{k+1} - x_k| < \varepsilon, \quad |x_{k+1} - x_k| < \varepsilon |x_k|, \quad f(x_k) - f(x_{k+1}) < \\ \varepsilon, \quad f(x_k) - f(x_{k+1}) < \varepsilon |f(x_k)|, \end{aligned}$$

kde ε je předepsaná tolerance.

Výpočet x_{k+1} , $k = 0, 1, \dots$, vyžaduje $k + 4$ vyhodnocení účelové funkce: $f(a_0)$, $f(c_0)$, $f(b_0)$ a dále $f(x_i)$, $i = 0, 1, \dots, k$.

Pokud metoda kvadratické interpolace konverguje, pak je rychlost její konvergence superlineární řádu $p \approx 1,324$, viz [Heath].

Příklad 6.2. Minimum funkce $f(x) = x^4 - 3x^3 + x + 7$ určíme metodou kvadratické interpolace. Výpočet ukončíme, když $|x_{k+1} - x_k| < 10^{-5}$. Výpočet je zaznamenán v následující tabulce. V posledním sloupci je uvedeno, který z případů A,B,C,D nastává, a podtržením

$a_0 = \underline{2,000000}$	$x_1 = \underline{2,204918}$	$<$	$c_0 = \underline{2,500000}$	$b_0 = 3,000000$	$f(x_1) < f(c_0)$
$a_1 = 2,000000$	$x_2 = \underline{2,180689}$	$<$	$c_1 = \underline{2,204918}$	$b_1 = \underline{2,500000}$	$f(x_2) \geq f(c_1)$
$a_2 = \underline{2,180689}$	$x_3 = \underline{2,197322}$	$<$	$c_2 = \underline{2,204918}$	$b_2 = 2,500000$	$f(x_3) < f(c_2)$
$a_3 = 2,180689$	$c_3 = \underline{2,197322}$	$<$	$x_4 = \underline{2,198232}$	$b_3 = \underline{2,204918}$	$f(c_3) \geq f(x_4)$
$a_4 = 2,197322$	$c_4 = \underline{2,198232}$	$<$	$x_5 = \underline{2,198264}$	$b_4 = \underline{2,204918}$	$f(c_4) \geq f(x_5)$
$a_5 = 2,198232$	$c_5 = 2,198264$	$<$	$x_6 = 2,198265$	$b_5 = 2,204918$	

jsou (postupně zleva doprava) vyznačeny body $a_{k+1} < c_{k+1} < b_{k+1}$, viz následující řádek. Z tabulky je zřejmé, že požadovaná přesnost byla dosažena pro $x_6 \doteq 2,19827$ (všechny cifry jsou platné). Hodnota účelové funkce se počítá celkem 9-krát. \square

Brentova metoda je kombinovaná metoda, která v sobě spojuje spolehlivost metody zlatého řezu a rychlou konvergenci metody kvadratické interpolace. Popis Brentovy metody lze najít např. v [Numerical Recipes], viz funkce `brent`, nebo v [Moler], viz funkce `fminbx`. Brentův algoritmus je také základem funkce `fminbnd` pro jednorozměrnou minimalizaci v MATLABu.

Obsah

- 6 Optimalizace
 - Úvod
 - Jednorozměrná minimalizace
 - Minimalizace funkce více proměnných
 - Literatura

Nelderova-Meadova metoda známá také jako **metoda simplexů** je populární metoda nepoužívající derivace účelové funkce. Patří mezi tzv. **komparativní metody**, což jsou metody, které hledají minimum účelové funkce f porovnáváním jejích hodnot v určitých vybraných bodech prostoru \mathbb{R}^n . V případě metody simplexů jsou vybranými body

vrcholy simplexu (pro $n = 2$ trojúhelníka, pro $n = 3$ čtyřstěnu).

Hlavní myšlenka jednoho kroku metody je jednoduchá: mezi vrcholy $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n$ simplexu vybereme nejhorší vrchol \mathbf{x}_w (v angličtině „worst“), v němž účelová funkce nabývá největší hodnotu, a nahradíme ho lepším vrcholem $\hat{\mathbf{x}}$, v němž je hodnota účelové funkce menší.

Vrchol $\hat{\mathbf{x}}$ hledáme na polopřímce, která vychází z nejhoršího vrcholu \mathbf{x}_w a prochází těžištěm $\bar{\mathbf{x}}$ zbývajících vrcholů. Nejlepší z nich označíme \mathbf{x}_b (v angličtině „best“), tj. \mathbf{x}_b je ten z vrcholů $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n$, v němž účelová funkce nabývá nejmenší hodnotu.

První pokus, jak vybrat $\hat{\mathbf{x}}$, označujeme jako **reflexi**: bod $\mathbf{x}_r = \bar{\mathbf{x}} + (\bar{\mathbf{x}} - \mathbf{x}_w)$, je obraz bodu \mathbf{x}_w ve středové souměrnosti se středem $\bar{\mathbf{x}}$. Když je $f(\mathbf{x}_r) < f(\mathbf{x}_b)$, pak to znamená, že pokles hodnot na polopřímce $\mathbf{x}_w\bar{\mathbf{x}}$ je značný, a proto zkusíme postoupit po této polopřímce ještě dál, do bodu $\mathbf{x}_e = \bar{\mathbf{x}} + 2(\bar{\mathbf{x}} - \mathbf{x}_w)$. Výběr bodu \mathbf{x}_e bývá označován jako **expanze**. Když $f(\mathbf{x}_e) < f(\mathbf{x}_b)$, pak $\hat{\mathbf{x}} = \mathbf{x}_e$, tj. nejhorší vrchol \mathbf{x}_w nahradíme bodem \mathbf{x}_e .

Jestliže $f(\mathbf{x}_e) \geq f(\mathbf{x}_b)$, zkusíme použít alespoň bod \mathbf{x}_r . Podmínkou pro jeho zařazení do simplexu je splnění podmínky $f(\mathbf{x}_r) < f(\mathbf{x}_g)$ pro některý vrchol \mathbf{x}_g jiný než nejhorší, tj. pro $\mathbf{x}_g \neq \mathbf{x}_w$ (index g připomíná anglické slůvko „good“). Pokud taková podmínka platí, bereme $\hat{\mathbf{x}} = \mathbf{x}_r$ místo původního \mathbf{x}_w .

Když nevyhovuje \mathbf{x}_e ani \mathbf{x}_r , zkusíme najít bod $\hat{\mathbf{x}}$ na úsečce s koncovými body \mathbf{x}_w , \mathbf{x}_r tak, aby $f(\hat{\mathbf{x}}) < \min\{f(\mathbf{x}_w); f(\mathbf{x}_r)\}$. Konkrétně postupujeme takto:

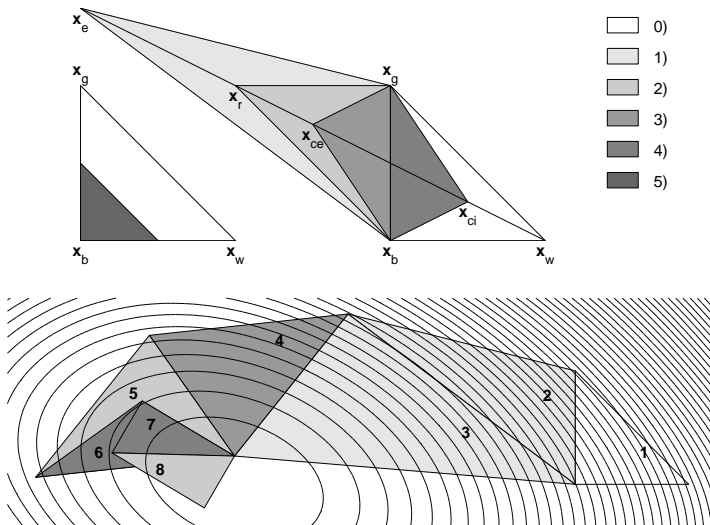
- a) pokud $f(\mathbf{x}_r) < f(\mathbf{x}_w)$, zkusíme bod $\mathbf{x}_{ce} = \frac{1}{2}(\bar{\mathbf{x}} + \mathbf{x}_r)$ (leží blíže k bodu \mathbf{x}_r),
a když $f(\mathbf{x}_{ce}) < f(\mathbf{x}_r)$, pak $\hat{\mathbf{x}} = \mathbf{x}_{ce}$, takže provedeme $\mathbf{x}_w := \mathbf{x}_{ce}$.
- b) jestliže $f(\mathbf{x}_r) \geq f(\mathbf{x}_w)$, zkusíme bod $\mathbf{x}_{ci} = \frac{1}{2}(\bar{\mathbf{x}} + \mathbf{x}_w)$ (leží blíže k bodu \mathbf{x}_w),
a pokud $f(\mathbf{x}_{ci}) < f(\mathbf{x}_w)$, pak $\hat{\mathbf{x}} = \mathbf{x}_{ci}$, tj. provedeme $\mathbf{x}_w := \mathbf{x}_{ci}$.

Výběr bodu \mathbf{x}_{ce} resp. \mathbf{x}_{ci} označujeme jako **kontrakci** (v dolním indexu: písmeno c připomíná anglické slovo „contraction“, písmeno e anglické slovo „external“ (\mathbf{x}_{ce} leží vně původního simplexu) a písmeno i připomíná anglické slovo „internal“ (\mathbf{x}_{ci} leží uvnitř původního simplexu)).

Když nevyhovuje \mathbf{x}_e , \mathbf{x}_r , \mathbf{x}_{ce} ani \mathbf{x}_{ci} , usoudíme, že vrchol \mathbf{x}_b , v němž účelová funkce nabývá své nejmenší hodnoty, je blízko minima. Proto provedeme **redukci simplexu**: vrchol \mathbf{x}_b v simplexu zůstane a zbývající vrcholy \mathbf{x}_i se posunou do středu úseček $\mathbf{x}_b\mathbf{x}_i$. Simplex se tedy stáhne k nejlepšímu vrcholu \mathbf{x}_b .

Transformaci simplexu, představující jeden krok Nelderovy Meadovy metody, popíšeme v pěti bodech takto:

- 1) expanze : $f(\mathbf{x}_r) < f(\mathbf{x}_b)$ a navíc $f(\mathbf{x}_e) < f(\mathbf{x}_b) \implies \mathbf{x}_w := \mathbf{x}_e$
- 2) reflexe : $f(\mathbf{x}_r) < f(\mathbf{x}_g)$ pro nějaký vrchol $\mathbf{x}_g \neq \mathbf{x}_w \implies \mathbf{x}_w := \mathbf{x}_r$
- 3) vnější kontrakce : $f(\mathbf{x}_r) < f(\mathbf{x}_w)$ a navíc $f(\mathbf{x}_{ce}) < f(\mathbf{x}_r) \implies \mathbf{x}_w := \mathbf{x}_{ce}$
- 4) vnitřní kontrakce : $f(\mathbf{x}_r) \geq f(\mathbf{x}_w)$ a navíc $f(\mathbf{x}_{ci}) < f(\mathbf{x}_w) \implies \mathbf{x}_w := \mathbf{x}_{ci}$
- 5) redukce : $\mathbf{x}_i := \frac{1}{2}(\mathbf{x}_b + \mathbf{x}_i)$ pro všechna $\mathbf{x}_i \neq \mathbf{x}_b$



Obr. 6.4: Nelderova-Meadova metoda: 0) originální trojúhelník, 1) expanze, 2) reflexe, 3) vnější kontrakce, 4) vnitřní kontrakce, 5) redukce

Body 1 až 5 procházíme postupně shora dolů. Když některá z podmínek v bodech 1 až 4 není splněna, přejdeme na následující bod. Když splněna je, provedeme náhradu \mathbf{x}_w podle příkazu za šipkou a transformace je hotova. Není-li splněna podmínka v žádném z bodů 1 až 4, provedeme redukci podle bodu 5.

Na začátku výpočtu je dána počáteční aproximace $\mathbf{x}_0 = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^T$ a malé číslo δ . Další vrcholy \mathbf{x}_i **startovacího simplexu** odvodíme z vrcholu \mathbf{x}_0 tak, že k jeho i -té složce $x_i^{(0)}$ přičteme číslo δ , tj. $\mathbf{x}_i = (x_1^{(0)}, \dots, x_i^{(0)} + \delta, \dots, x_n^{(0)})^T$, $i = 1, 2, \dots, n$.

Simplex opakovaně transformujeme. Výpočet ukončíme a vrchol \mathbf{x}_b považujeme za dostatečně dobrou aproximaci minima \mathbf{x}^* , pokud jsou vrcholy simplexu navzájem dosti blízko a funkční hodnoty v nich se málo liší, tj. když pro zadané tolerance ε_1 , ε_2 platí

$$\|\mathbf{x}_i - \mathbf{x}_b\| < \varepsilon_1 \quad \text{a současně} \quad |f(\mathbf{x}_i) - f(\mathbf{x}_b)| < \varepsilon_2 \quad \text{pro každý vrchol } \mathbf{x}_i \neq \mathbf{x}_b. \quad (6.6)$$

Nelderova-Meadova metoda je **heuristická metoda** (heuristický nebo-li objevovací je takový postup, který je založen nejenom na logickém uvažování a zkušenostech, ale také na pozorování a experimentování). Metoda je vhodná pro minimalizaci funkcí menšího počtu proměnných, řekněme pro $n \leq 10$. Přestože o konvergenci metody je toho známo jen velmi málo, praxe hovoří v její prospěch: metoda je až překvapivě úspěšná. Proto je považována za metodu spolehlivou nebo-li **robustní**. Podstatnou nevýhodou metody je to, že je pomalá, zejména v blízkosti minima. Další mínus představuje velký objem výpočtů. Přesto nejde o „mrtvou metodu“, což je zřejmé např. z toho, že je implementována jako základní metoda vícerozměrné minimalizace v programu `fminsearch` jádra MATLABu.

Příklad 6.3. Funkce $f(x, y) = 70 [(x - 2)^4 + (x - 2y)^2]$ nabývá minima pro $x^* = 2$, $y^* = 1$. Výpočet provedeme metodou Nelder-Meada, v níž zvolíme $\mathbf{x}_0 = (2, 1; 0, 7)^T$ a $\delta = 0,1$. Výpočty jsou prováděny přesně, do tabulky 6.1 však (kvůli úspoře místa) zapisujeme hodnoty zaokrouhlené na dvě desetinná místa. Pro každý bod zapisujeme ve sloupci pod sebou x -ovou souřadnici, y -ovou souřadnici a funkční hodnotu. V rámečku uvádíme nově určený lepší bod $\hat{\mathbf{x}}$ (a funkční hodnotu v něm). Pro $\varepsilon_1 = 10^{-4}$ a $\varepsilon_2 = 10^{-8}$ výpočet končí po 47 krocích a $\mathbf{x}_b \doteq (1,999953; 0,999976)^T$. \square

krok	typ	x_w	x_g	x_b	\bar{x}	x_r	x_e	x_{ce}	x_{ci}
1	expanze	2,20	2,10	2,10	2,10	2,00	1,90		
		0,70	0,70	0,80	0,75	0,80	0,85		
		44,91	34,31	17,51	—	11,20	2,81		
2	reflexe	2,10	2,10	1,90	2,00	1,90	1,80		
		0,70	0,80	0,85	0,82	0,95	1,08		
		34,31	17,51	2,81	—	0,01	8,69		
3	vnější kontrakce	2,10	1,90	1,90	1,90	1,70		1,80	
		0,80	0,85	0,95	0,90	1,00		0,95	
		17,51	2,81	0,01	—	6,87		0,81	
4	vnitřní kontrakce	1,90	1,80	1,90	1,85	1,80			1,88
		0,85	0,95	0,95	0,95	1,05			0,90
		2,81	0,81	0,01	—	6,41			0,41

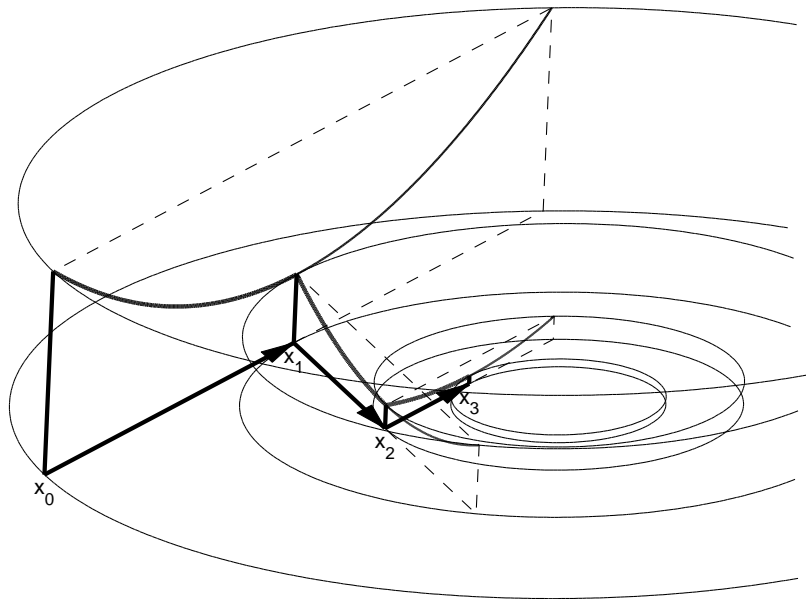
Tabulka 6.1: Příklad 6.3

Metoda největšího spádu je základní minimalizační metoda, která používá derivace účelové funkce. Takové **metody** se nazývají **gradientní**. Začneme tím, že si vysvětlíme obecný princip **spádové metody**. Předpokládejme tedy, že jsme v bodu \mathbf{x}_k a chceme se dostat blíže k minimu. Zvolíme směr \mathbf{d}_k , v němž funkce f klesá, a na polopřímce $\mathbf{x}_k + \lambda \mathbf{d}_k$, $\lambda \geq 0$, vybereme bod

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k, \quad (6.7)$$

v němž $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$. **Směrový vektor** \mathbf{d}_k nazýváme **spádový** (ve směru \mathbf{d}_k hodnota účelové funkce f „padá dolů“), odtud spádová metoda. Číslo λ_k se nazývá **parametr délky kroku** (je-li \mathbf{d}_k jednotkový vektor, tj. když $\|\mathbf{d}_k\|_2 = 1$, pak $\lambda_k = \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2$ je vzdálenost bodů \mathbf{x}_k a \mathbf{x}_{k+1} , tj. λ_k je délka kroku). λ_k dostaneme minimalizací funkce $\varphi(\lambda) = f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ pro $\lambda \geq 0$. Minimum λ_k funkce $\varphi(\lambda)$ určíme přibližně pomocí několika málo kroků vhodné metody jednorozměrné minimalizace (přesná minimalizace je zbytečný přepych, neboť prostřednictvím λ_k určujeme jen mezivýsledek \mathbf{x}_{k+1} na cestě k minimu \mathbf{x}^*). Určení λ_k tedy vyjádříme zápisem

$$\lambda_k \approx \min_{\lambda > 0} \varphi(\lambda), \quad \text{kde} \quad \varphi(\lambda) = f(\mathbf{x}_k + \lambda \mathbf{d}_k). \quad (6.8)$$



Obr. 6.5: Princip spádových metod

Nyní se věnujme už vlastní metodě největšího spádu. Je známo, že funkce $f(\mathbf{x})$ nejrychleji klesá ve směru záporného gradientu. Označíme-li tedy gradient jako

$$\mathbf{g}(\mathbf{x}) \equiv \nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)^T,$$

pak v metodě největšího spádu volíme jako spádový vektor

$$\mathbf{d}_k = -\mathbf{g}(\mathbf{x}_k). \quad (6.9)$$

Výpočet ukončíme a \mathbf{x}_{k+1} považujeme za uspokojivou aproximaci minima \mathbf{x}^* , když

$$\|\mathbf{g}(\mathbf{x}_{k+1})\| < \varepsilon, \quad (6.10)$$

kde ε je předepsaná tolerance (připomeňme, že v minimu $\mathbf{g}(\mathbf{x}^*) = \mathbf{o}$).

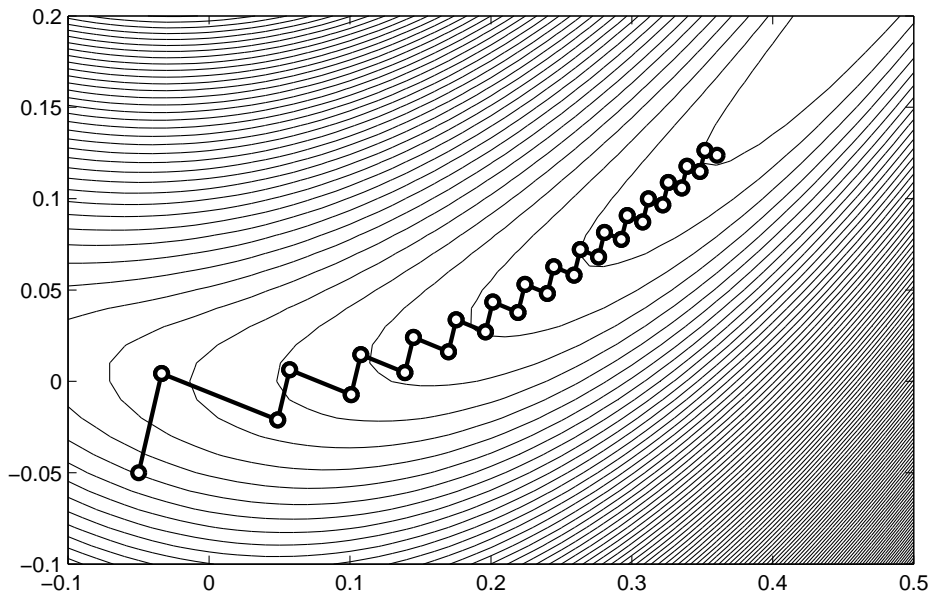
V počáteční fázi výpočtu, když jsme od minima ještě dosti daleko, dochází obvykle k poměrně rychlému poklesu hodnot účelové funkce. Zato v blízkosti minima je konvergence pomalá, jen lineární, tj. platí $\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq C \|\mathbf{x}_k - \mathbf{x}^*\|$, kde konstanta C je sice menší než jedna, ale často jen nepatrně.

Cik-cak efekt. Pokud λ_k vypočteme přesně, má funkce $\varphi(\lambda)$ v bodě λ_k minimum, a proto

$$0 = \varphi'(\lambda_k) = \sum_{i=1}^n \frac{\partial f(\mathbf{x}_{k+1})}{\partial x_i} d_i^{(k)} = - \sum_{i=1}^n d_i^{(k+1)} d_i^{(k)} = -\mathbf{d}_{k+1}^T \mathbf{d}_k,$$

tj. směrové vektory \mathbf{d}_{k+1} a \mathbf{d}_k jsou navzájem kolmé. Ukážeme si, že právě tato vlastnost může být příčinou velmi pomalé konvergence.

Pro jednoduchost předpokládejme, že minimalizujeme funkci dvou proměnných. V tom případě si můžeme vypomoci jednoduchou představou: nacházíme se v terénu a chceme najít nejnižší bod, tj. dno nějaké prohlubně. Předpokládejme, že prohlubeň má tvar **podlouhlé zahnuté rokle**. Metoda největšího spádu nás nasměruje z počátečního stanoviště \mathbf{x}_0 dolů kolmo k vrstevnici $f(\mathbf{x}) = f(\mathbf{x}_0)$. Sestupujeme tak dlouho, dokud terén klesá. V nejnižším místě je další stanoviště \mathbf{x}_1 . Zde se zastavíme, otočíme se o 90 stupňů a sestupujeme znovu dolů (kolmo k vrstevnici $f(\mathbf{x}) = f(\mathbf{x}_1)$) do dalšího stanoviště \mathbf{x}_2 atd. Je zřejmé, že vzdálenosti mezi jednotlivými stanovišti se budou postupně zkracovat. V blízkosti minima bude naše putování působit směšně: místo abychom do něj došli několika málo kroky, budeme se k němu spíše plížit než blížít po trase tvořené čím dál tím kratšími navzájem kolmými úseky. Tento jev bývá označován jako **cik-cak efekt**. V takovém extrémním případě metoda největšího spádu selhává, neboť počet kroků potřebný k dosažení přijatelné aproximace minima je neúnosně velký. Obdobná situace nastane i v případě, když jednorozměrnou minimalizaci provádíme jen přibližně. \square



Obr. 6.6: Cik-cak efekt

Příklad 6.4. Funkci $f(x, y) = x^2 + 2y^2 + xy - x$ minimalizujeme metodou největšího spádu. Jako počáteční aproximaci zvolíme $x_0 = 1$, $y_0 = 2$. Podrobně provedeme první dva kroky. Nejdříve určíme

$$\mathbf{g}(x, y) = \begin{pmatrix} 2x + y - 1 \\ x + 4y \end{pmatrix} \quad \text{a odtud} \quad \mathbf{d}(x, y) = \begin{pmatrix} -2x - y + 1 \\ -x - 4y \end{pmatrix}.$$

Pro

$$\mathbf{x}_0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \text{tak dostaneme} \quad \mathbf{d}_0 = \begin{pmatrix} -3 \\ -9 \end{pmatrix}, \quad \text{takže} \quad \mathbf{x}_1 = \mathbf{x}_0 + \lambda_0 \mathbf{d}_0 = \begin{pmatrix} 1 - 3\lambda_0 \\ 2 - 9\lambda_0 \end{pmatrix}.$$

Funkce $\varphi(\lambda)$ je tedy tvaru

$$\begin{aligned} \varphi(\lambda) &= f(\mathbf{x}_0 + \lambda \mathbf{d}_0) = (1 - 3\lambda)^2 + 2(2 - 9\lambda)^2 + (1 - 3\lambda)(2 - 9\lambda) - (1 - 3\lambda) = \\ &= 198\lambda^2 - 90\lambda + 10. \end{aligned}$$

Minimalizaci funkce $\varphi(\lambda)$ umíme provést přesně:

$$\begin{aligned} \text{z podmínky } \varphi'(\lambda_0) = 396\lambda_0 - 90 = 0 \text{ dostaneme } \lambda_0 = \frac{5}{22}, \text{ takže} \\ \mathbf{x}_1 = \begin{pmatrix} 1 - 3 \cdot 5/22 \\ 2 - 9 \cdot 5/22 \end{pmatrix} = \frac{1}{22} \begin{pmatrix} 7 \\ -1 \end{pmatrix} \doteq \begin{pmatrix} 0,318 \\ -0,045 \end{pmatrix}. \end{aligned}$$

Další krok provádíme podobně. Nejdříve určíme \mathbf{d}_1 a pak zapíšeme hledaný tvar pro \mathbf{x}_2 :

$$\mathbf{d}_1 = \begin{pmatrix} -2 \cdot 7/22 - (-1/22) + 1 \\ -7/22 - 4 \cdot (-1/22) \end{pmatrix} = \frac{1}{22} \begin{pmatrix} 9 \\ -3 \end{pmatrix}, \quad \mathbf{x}_2 = \mathbf{x}_1 + \lambda_1 \mathbf{d}_1 = \frac{1}{22} \begin{pmatrix} 7 + 9\lambda_1 \\ -1 - 3\lambda_1 \end{pmatrix}$$

Odtud pro $\varphi(\lambda) = f(\mathbf{x}_1 + \lambda \mathbf{d}_1)$ po úpravě dostaneme

$$\varphi(\lambda) = \frac{1}{22^2} (72\lambda^2 - 90\lambda - 110) \text{ a z podmínky } \varphi'(\lambda_1) = 0 \text{ máme } \lambda_1 = \frac{5}{8}.$$

Dosazením za λ_1 do \mathbf{x}_2 nakonec obdržíme

$$\mathbf{x}_2 = \frac{1}{176} \begin{pmatrix} 101 \\ -23 \end{pmatrix} \doteq \begin{pmatrix} 0,574 \\ -0,131 \end{pmatrix}.$$

V tomto ilustračním příkladu metoda největšího spádu rychle konverguje k přesnému řešení $x^* = 4/7$, $y^* = -1/7$: už hodnoty x_6 , y_6 jsou na 6 desetinných míst přesné. \square

Poznámka (**O spádových metodách pro řešení soustav lineárních rovnic**). Když \mathbf{A} je pozitivně definitní matice, pak pro funkci $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}$ je $\mathbf{g}(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b}$ a $\mathbf{g}'(\mathbf{x}) = \mathbf{A}$, takže jediné minimum \mathbf{x}^* funkce $f(\mathbf{x})$ je řešením soustavy lineárních rovnic $\mathbf{A} \mathbf{x} = \mathbf{b}$. Pro spádový vektor \mathbf{d}_k počítáme $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$, přičemž délku λ_k kroku vyjádříme z podmínky $0 = \varphi'(\lambda_k) = \frac{d}{d\lambda} f(\mathbf{x}_k + \lambda \mathbf{d}_k) \big|_{\lambda=\lambda_k} = \mathbf{d}_k^T \mathbf{A} \mathbf{d}_k \lambda_k - \mathbf{d}_k^T \mathbf{r}_k$, kde $\mathbf{r}_k = \mathbf{b} - \mathbf{A} \mathbf{x}_k$. (Ověřte!) Tak dostaneme metodu: \mathbf{x}_0 dáno, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$, pro $k = 0, 1, \dots$ počítáme

$$\mathbf{v} = \mathbf{A} \mathbf{d}_k, \quad \lambda_k = \frac{\mathbf{d}_k^T \mathbf{r}_k}{\mathbf{d}_k^T \mathbf{v}}, \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k, \quad \mathbf{r}_{k+1} = \mathbf{r}_k - \lambda_k \mathbf{v}. \quad (6.11)$$

(Vzorec pro \mathbf{r}_{k+1} obdržíme tak, že do $\mathbf{r}_{k+1} = \mathbf{b} - \mathbf{A} \mathbf{x}_{k+1}$ dosadíme $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$.) Výpočet ukončíme, když je reziduum \mathbf{r}_k dostatečně malé, tj. když $\|\mathbf{r}_{k+1}\| < \varepsilon$. V metodě největšího spádu $\mathbf{d}_k = -\mathbf{g}'(\mathbf{x}_k) = \mathbf{r}_k$. V praxi se však tato metoda nepoužívá, neboť obvykle konverguje příliš pomalu. Existuje lepší volba: zvolíme-li $\mathbf{d}_0 = \mathbf{r}_0$, a počítáme-li \mathbf{d}_{k+1} pro $k > 0$ ze vzorce $\mathbf{d}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k$, v němž $\beta_k = (\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}) / (\mathbf{r}_k^T \mathbf{r}_k)$, dostaneme **metodu sdružených gradientů**, která konverguje podstatně rychleji, viz [Meurant], [Stoer, Bulirsch], [Heath].

Newtonova metoda je gradientní metoda, která se pokouší najít minimum jako řešení \mathbf{x}^* soustavy nelineárních rovnic $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ Newtonovou metodou. Takovým řešením ale může být každý stacionární bod funkce f (bod \mathbf{x}^* , ve kterém pro funkci f platí $\nabla f(\mathbf{x}^*) = 0$, se nazývá **stacionární bod** funkce f), tedy také maximum nebo sedlový bod (obdobu inflexního bodu pro funkci jedné proměnné). Budeme tedy potřebovat Jacobiovu matici

$$\mathbf{g}'(\mathbf{x}) \equiv \mathbf{H}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{pmatrix},$$

kteřá se v tomto případě nazývá **Hessova matice** funkce f .

Výpočet provádíme podle vzorců z kapitoly *řešení nelineárních rovnic*, tj. najdeme řešení \mathbf{d}_k lineární soustavy rovnic

$$\mathbf{H}(\mathbf{x}_k) \mathbf{d}_k = -\mathbf{g}(\mathbf{x}_k) \quad \text{a pak určíme} \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k. \quad (6.12)$$

Všimněte si, že vzorec (6.12) pro \mathbf{x}_{k+1} je formálně stejný jako vzorec (6.7), když v něm $\lambda_k = 1$. Pro ukončení výpočtu použijeme stop kritérium (6.10). Směrový vektor

$$\mathbf{d}_k = -\mathbf{H}^{-1}(\mathbf{x}_k) \mathbf{g}(\mathbf{x}_k) \quad (6.13)$$

obecně není spádový. Dá se však ukázat, že v blízkosti minima vektor \mathbf{d}_k spádový je.

Z kapitoly *řešení nelineárních rovnic* už víme, že Newtonova metoda konverguje, když je počáteční aproximace \mathbf{x}_0 dostatečně blízko řešení \mathbf{x}^* , a že v tom případě je konvergence rychlá, totiž kvadratická. Na druhé straně, když \mathbf{x}_0 není dosti blízko \mathbf{x}^* , Newtonova metoda vůbec konvergovat nemusí. Proto se nabízí možnost začít výpočet metodou největšího spádu a dokončit ho Newtonovou metodou. Tento poznatek je východiskem pro odvození řady efektivních metod. Mezi takové patří například

Kvazinewtonovské metody. Výpočet probíhá podobně jako v metodě největšího spádu. Směrový vektor \mathbf{d}_k je však tvaru $\mathbf{d}_k = -\mathbf{B}_k \mathbf{g}(\mathbf{x}_k)$. Matice \mathbf{B}_k se vybírají tak, aby vektor \mathbf{d}_k byl spádový. Pro $k = 0$ je přitom $\mathbf{B}_0 = \mathbf{I}$ jednotková matice, takže $\mathbf{d}_0 = -\mathbf{g}(\mathbf{x}_0)$ je směr metody největšího spádu. Čím víc se \mathbf{x}_k blíží minimu, tím je \mathbf{B}_k lepší aproximací $\mathbf{H}^{-1}(\mathbf{x}_k)$, takže \mathbf{d}_k se blíží směrovému vektoru $-\mathbf{H}^{-1}(\mathbf{x}_k) \mathbf{g}(\mathbf{x}_k)$ Newtonovy metody. To je jen velmi hrubý popis, detaily viz např. [Nocedal], [Stoer, Bulirsch].

Příklad 6.5. Pro testování kvality minimalizačních metod se používá mezi jinými také tzv. „banánová funkce“ $f(x, y) = 100(y - x^2)^2 + (1 - x)^2$. Průběh funkce f navozuje představu hluboké rokle se strmými stěnami, jejíž zakřivené dno o rovnici $y = x^2$ (vzdáleně připomínající banán) se jen velmi mírně svažuje k minimu v bodě $x^* = y^* = 1$.

Minimum hledejme Newtonovou metodou, tj. soustavu rovnic

$$\frac{\partial f(x, y)}{\partial x} = -400x(y - x^2) - 2(1 - x) = 0, \quad \frac{\partial f(x, y)}{\partial y} = 200(y - x^2) = 0$$

řešíme Newtonovou metodou: v každém kroku vypočteme a_k, b_k ze soustavy rovnic

$$\begin{pmatrix} 1200x_k^2 - 400y_k + 2 & -400x_k \\ -400x_k & 200 \end{pmatrix} \begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} 400x_k(y_k - x_k^2) + 2(1 - x_k) \\ -200(y_k - x_k^2) \end{pmatrix}$$

a potom určíme další aproximaci

$$x_{k+1} = x_k + a_k, \quad y_{k+1} = y_k + b_k.$$

Pro numerický experiment jsme zvolili počáteční aproximaci $x_0 = 3$, $y_0 = 2$. Výpočet potvrdil velmi rychlou konvergenci Newtonovy metody, neboť už v pátém kroku jsme dostali aproximaci, která měla 16 platných cifer. To je skvělé!










Pro srovnání jsme tutéž úlohu řešili také metodou největšího spádu. Jednorozměrnou minimalizaci (6.8) jsme provedli přibližně metodou zlatého řezu na intervalu $\langle 0, 1 \rangle$ s přesností 10^{-4} . Pro $\varepsilon = 10^{-3}$ byla podmínka (6.10) splněna až pro $k = 1324$, kdy jsme dostali aproximaci $(1,001; 1,001)^T$. To je neuspokojivé.









To metoda Nelder-Meada si vedla lépe. Pro výpočet jsme vybrali parametry $\delta = 0,2$, $\varepsilon_1 = 10^{-2}$ a $\varepsilon_2 = 10^{-4}$. Podmínka (6.6) byla splněna po 57 krocích, kdy jsme dostali aproximaci $(1,001; 1,001)^T$. To je přijatelné. \square







Poznámka. Řešení \mathbf{x}^* soustavy n nelineárních rovnic $\mathbf{f}(\mathbf{x}) = \mathbf{o}$ můžeme získat jako bod, v němž funkce $h(\mathbf{x}) = \sum_{i=1}^n [f_i(\mathbf{x})]^2$ (tj. součet čtverců reziduí) nabývá svého globálního minima (neboť $0 = h(\mathbf{x}^*) = \min_{\mathbf{x} \in \mathbf{R}^n} h(\mathbf{x})$, právě když $\mathbf{f}(\mathbf{x}^*) = \mathbf{o}$). Minimalizaci $h(\mathbf{x})$ lze provádět pomocí algoritmů pro řešení nelineární úlohy nejmenších čtverců, viz např. [Nocedal].

Obsah

- 6 Optimalizace
 - Úvod
 - Jednorozměrná minimalizace
 - Minimalizace funkce více proměnných
 - Literatura

-  I.S. Berezin, N.P. Židkov: *Číslennyje metody I,II*, Nauka, Moskva, 1962.
-  G. Dahlquist, G. Å Björk: *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
-  M. Fiedler: *Speciální matice a jejich použití v numerické matematice*, SNTL, Praha, 1981.
-  D. Hanselman, B. Littlefield: *Mastering MATLAB 7*, Pearson Prentice Hall, Upper Saddle River, NJ, 2005.
-  G. Hämmerlin, K. H. Hoffmann: *Numerical Mathematics*, Springer-Verlag, Berlin, 1991.
-  M. T. Heath: *Scientific Computing. An Introductory Survey*, McGraw-Hill, New York, 2002.
-  I. Horová, J. Zelinka: *Numerické metody*, učební text Masarykovy univerzity, Brno, 2004.
-  J. Kobza: *Splajny*, učební text Palackého univerzity, Olomouc, 1993.
-  J. Klapka, J. Dvořák, P. Popela: *Metody operačního výzkumu*, učební text, FSI VUT Brno, 2001.

-  J. H. Mathews, K. D. Fink: *Numerical Methods Using MATLAB*, Pearson Prentice Hall, New Jersey, 2004.
-  MATLAB: *Mathematics*, Version 7, The MathWorks, Inc., Natick, 2004.
-  G. Meurant: *Computer Solution of Large Linear Systems*, Elsevier, Amsterdam, 1999.
-  S. Míka: *Numerické metody algebry*, SNTL, Praha, 1985..
-  C. B. Moler: *Numerical Computing with MATLAB*, Siam, Philadelphia, 2004.
<http://www.mathworks.com/moler>.
-  J. Nocedal, S. J. Wright: *Numerical Optimization, Springer Series in Operations Research*, Springer, Berlin, 1999.
-  A. Quarteroni, R. Sacco, F. Saleri: *Numerical Mathematics*, Springer, Berlin, 2000.
-  W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling: *Numerical Recipes in Pascal, The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1990.

-  P. Přikryl: *Numerické metody matematické analýzy*, SNTL, Praha, 1985.
-  A. R. Ralston: *Základy numerické matematiky*, Academia, Praha, 1973.
-  K. Rektorys: *Přehled užití matematiky I,II*, Prometheus, Praha, 1995.
-  J. Stoer, R. Bulirsch: *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1993.
-  E. Vitásek: *Numerické metody*, SNTL, Praha, 1987.
-  W. Y. Yang, W. Cao, T. S. Chung, J. Morris: *Applied Numerical Methods Using Matlab*, John Willey & Sons, New Jersey, 2005.