FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITY KOMENSKÉHO V BRATISLAVE

Diplomová práca

Bratislava 2007

Zuzana Hasonová

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITY KOMENSKÉHO V BRATISLAVE

Ekonomická a finančná matematika

Algoritmy primárno-duálnych metód vnútorného bodu na riešenie úloh lineárneho programovania

Diplomová práca

Diplomant: Zuzana Hasonová

Vedúci diplomovej práce: Doc. RNDr. Milan Hamala, CSc.

Bratislava 2007

Čestne prehlasujem, že som diplomovú prácu vypracovala samostatne a s použitím uvedenej literatúry.

....

Ďakujem vedúcemu mojej diplomovej práce p. Doc. RNDr. Milanovi Hamalovi, CSc., za umožnenie odovzdania práce. Úprimne ďakujem aj svojím rodičom za všestrannú podporu počas celého štúdia.

Obsah

Úvod

Úloha lineárnej optimalizácie datuje svoje prvé zmienky v 30. a 40. rokoch 20. storočia. Hlavným štartérom zväčšujúceho záujmu o ňu boli vzrastajúce potreby o operačný výskum pre armádne účely v čase vojen, ale aj pre národné hospodárstva a mnohé oblasti vedy a výskumu. Novú éru samostatného vývoja lineárneho programovania (LP) odštartoval v roku 1947 George B. Dantzig svojim prevratným článkom, v ktorom navrhol použitie simplexového algoritmu na riešenie úloh LP. Simplexová metóda bola dlho považovaná za univerzálnu a spoľahlivú a iné prístupy riešenia ostávali mimo záujmu. Zlom nastal začiatkom 70-tych rokov, kedy V. Klee a G. Minty odvodili triedu príkladov, ktoré popreli polynomiálnu zložitosť simplexovej metódy. Tento úžasný objav dal priestor pokusom o alternatívne riešenie úloh lineárneho programovania.

Bariérové metódy boli spočiatku rozpracované v štruktúre nelineárneho programovania (NLP). Aj keď boli aplikovateľné na úlohy LP, nepovažovali ich za schopného konkurenta simplexového algoritmu. Prvé aplikácie logaritmickej bariérovej metódy sa pripisujú K. R. Frischovi (rok 1955). O viac ako desať rokov neskôr A. V. Fiacco a G. P. McCormick rozvinuli bariérovú metódu pre úlohy konvexného programovania. V roku 1979 L. G. Khachiyan síce odvodil elipsoidný algoritmus pre LP a dokázal jeho polynomialitu, v praxi však bol podstatne horší ako simplexová metóda.

Revolúcia nastala až po roku 1984, kedy N. Karmarkar publikoval projektívny algoritmus, ktorý vykazoval polynomiálnu zložitosť. Navyše pre úlohy veľkých rozmerov bol aj oveľa efektívnejší v porovnaní so simplexovou metódou. Ako sa ukázalo, tento algoritmus úzko súvisel s logaritmickou bariérovou metódou vnútorného bodu používanou v NLP. Tento objav následne spustil obrovský prúd záujmu výskumníkov o túto tému. V teórii LP sa začal rozvíjať nový smer riešenia úloh – metódy vnútorného bodu.

Dnes sa v moderných metódach vnútorného bodu používa široká škála primárno-duálnych algoritmov spadajúcich do súborov metód sledujúcich centrálnu trajektóriu, afinne-škálovacích metód či redukcie-schopných metód.

Cieľom mojej diplomovej práce bolo stručne popísať známe primárno-duálne metódy vnútorného bodu riešenia úloh lineárneho programovania a realizáciou numerického experimentu porovnať „klasické“ metódy (logaritmickú a nelogaritmickú) s metódou z „novšej“ triedy metód.

Diplomová práca je rozdelená do 4 kapitol. Prvá kapitola sa týka základných pojmov a poznatkov z teórie lineárneho programovania. Kapitola druhá sa stručne zaoberá základmi metód vnútorného bodu. V tretej kapitole sa venujeme konkrétnym vybraným metódam riešenia úloh LP. Štvrtá kapitola obsahuje popis algoritmov použitých metód a výsledky numerického experimentu. Zdrojový kód programu a tabuľkové výstupy sú uvedené v prílohách.

1. Teória lineárneho programovania

1.1. Úloha lineárneho programovania

Cieľom lineárneho programovania je minimalizovať lineárnu účelovú funkciu viacerých premenných viazanú lineárnymi podmienkami v tvare rovníc alebo nerovníc. Zavedieme si nasledovné označenie: $A = (a_{ij})$ je matica z $R^{m \times n}$, $c = (c_j)$ a $x = (x_j)$ sú vektory z R^n a $b = (b_i)$, $y = (y_i)$ sú vektory R^m, pričom vektory považujeme za stĺpce. Transponovaný vektor k vektoru $x \in R^n$ (riadkový vektor) označujeme x^T, podobne transponovanú maticu k matici A značíme A^T. Skalárny súčin dvoch vektorov c, x zapisujeme v tvare $c^T x$.

Úlohou lineárnej optimalizácie je nájsť minimum lineárnej účelovej funkcie n-premenných

$$c^T x = \sum_{j=1}^{n} c_j x_j$$

na množine všetkých nezáporných riešení sústavy m-lineárnych rovníc

$$Ax = b, \;\; x \geq 0_n,$$

$$\text{resp. } \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, \ldots, m,$$

$$x_j \geq 0, \quad j = 1, \ldots, n.$$

Symbolicky môžeme takto naformulovanú úlohu zapísať v tvare:

$$Min\left\{ c^T x \,\middle|\, Ax = b, x \geq 0_n \right\}.$$

Nazývame ju úloha lineárneho programovania v štandardnom tvare, skrátene štandardná úloha. Iné pomenovanie tejto úlohy môže byť úloha lineárneho programovania v rovnicovom tvare. V prípade, že hľadáme minimum lineárnej funkcie pri ohraničeniach v tvare nerovníc, riešime úlohu lineárneho programovania v tvare nerovností. Ľahko ju však vieme previesť na štandardnú úlohu pridaním doplnkových premenných $w_i \geq 0, \;\; i = 1, \ldots, m$ a úpravou nerovníc

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \;\; i = 1, \ldots, m$$

na rovnice nasledovne

$$\sum_{j=1}^{n} a_{ij} x_j + w_i = b_i, \;\; i = 1, \ldots, m.$$

Obdobne nahradením každej rovnice

$$\sum_{j=1}^{n} a_{ij} x_j = b_i$$

dvojicou nerovníc

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad \sum_{j=1}^{n} a_{ij} x_j \ge b_i$$

vieme každú úlohu v rovnicovom tvare prepísať na úlohu v tvare nerovností. Problém môže byť sformulovaný aj vo forme, kde namiesto minima budeme hľadať maximum danej účelovej funkcie. Pre ľubovoľnú funkciu $f : M \to R$, kde M je ľubovoľná množina $M \subset R^n$ platí:

$$Min(f(x)) = -Max(-f(x)), \quad x \in M ,$$

vďaka čomu možno každú maximalizačnú úlohu previesť na úlohu minimalizačnú a opačne. Ďalšou možnosťou je úloha obsahujúca voľné premenné. V štandardnej úlohe vyžadujeme nezápornosť všetkých premenných. Tú možno dosiahnuť rozdelením voľných premenných (napr. a) na dve nové nezáporné premenné $a^+ \ge 0$ a $a^- \ge 0$, ktoré musia spĺňať $a = a^+ - a^-$.

Odteraz sa budeme zaoberať iba úlohami lineárnej optimalizácie v štandardnom tvare.

1.2. Duálna úloha

Uvažujme štandardnú úlohu lineárneho programovania

$$(P) \quad Min\left\{ c^T x \,\middle|\, Ax = b, x \ge 0_n \right\}. \tag{1.1}$$

Zvyčajne sa táto minimalizačná úloha nazýva primárna úloha. V teórii lineárneho programovania k nej možno jednoznačne vytvoriť prislúchajúcu maximalizačnú úlohu

$$Max\left\{ b^T y \,\middle|\, A^T y \le c \right\},$$

pričom túto budeme nazývať duálna úloha k úlohe primárnej. Pomocou doplnkových premenných $z_j \ge 0, \quad j = 1, \dots, n$ ju prepíšeme do štandardného tvaru nasledovne:

$$(D) \quad Max\left\{ b^T y \,\middle|\, A^T y + z = c, z \ge 0_n \right\}. \tag{1.2}$$

Duálna úloha k maximalizačnej (duálnej) úlohe bude práve primárna minimalizačná úloha. Dá sa preto zaviesť pojem dvojica navzájom duálnych úloh.

Pre primárnu úlohu (P) definujme množinu primárne prípustných riešení

$$K = \left\{ x \in R^n \,\middle|\, Ax = b, x \ge 0_n \right\}. \tag{1.3}$$

Primárne prípustný bod, v ktorom účelová funkcia $c^T x$ nadobúda minimum na množine K, nazývame optimálne riešenie úlohy (P). Množinu všetkých optimálnych riešení úlohy (P) zapisujeme

$$\hat{K} = \left\{ \hat{x} \in K \,\middle|\, c^T \hat{x} \leq c^T x,\ \text{pre}\ \forall x \in K \right\}. \tag{1.4}$$

Obdobne pre duálnu úlohu (D) zadefinujeme množinu duálne prípustných riešení:

$$L = \left\{ (y, z) \in R^m \times R^n \,\middle|\, A^T y + z = c,\ z \geq 0_n \right\}. \tag{1.5}$$

Duálne prípustné riešenie, v ktorom účelová funkcia $b^T y$ nadobúda maximum na množine L, nazývame optimálne riešenie úlohy (D). Množinu všetkých optimálnych riešení úlohy (D) značíme

$$\hat{L} = \left\{ (\hat{y}, \hat{z}) \in L \,\middle|\, b^T \hat{y} \geq b^T y,\ \text{pre}\ \forall (y, z) \in L \right\}. \tag{1.6}$$

1.3. Základné vzťahy medzi primárnou a duálnou úlohou

Pri hľadaní optimálneho riešenia dvojice úloh (P) a (D) využívame nasledovné dôležité tvrdenia vyplývajúce z teórie duality.

Veta 1.3.1. (Základná veta lineárneho programovania)

Pre každú úlohu lineárneho programovania nastáva práve jedna z možností: neprípustnosť, neohraničenosť, optimalita.

Neprípustná úloha má prázdnu množinu prípustných riešení. Neohraničenosť nastáva, ak je neprázdna množina prípustných riešení úlohy neohraničená v smere optimalizácie účelovej funkcie. Optimalita je stav, keď množina optimálnych riešení úlohy nie je prázdna.

Veta 1.3.2. (Slabá veta o dualite)

Pre každé prípustné riešenie $x \in K$ primárnej úlohy (P) a každé prípustné riešenie $(y, z) \in L$ duálnej úlohy (D) platí: $c^T x \geq b^T y$.

Dôsledok 1.3.3.

Ak pre prípustné riešenie $x \in K$ primárnej úlohy (P) a prípustné riešenie $(y, z) \in L$ duálnej úlohy (D) platí rovnosť $c^T x = b^T y$, potom x je optimálnym riešením úlohy (P) a (y, z) je optimálnym riešením úlohy (D).

Rozdiel funkčných hodnôt účelových funkcií $c^T x - b^T y \geq 0$ sa nazýva duálna medzera a jej veľkosť je rozhodujúca pre overovanie optimality prípustných riešení (P) a (D). Ak $\hat{x}$ je prípustné riešenie primárnej úlohy (P) a $(\hat{y}, \hat{z})$ je prípustné riešenie duálnej úlohy (D) a veľkosť duálnej medzery pre ne je rovná nule $c^T \hat{x} - b^T \hat{y} = 0$, tak podľa dôsledku 1 $\hat{x}$, $(\hat{y}, \hat{z})$ sú aj optimálne.

Dôsledok 1.3.4.

Ak v jednej z úloh (P), (D) nastáva neohraničenosť, tak v druhej z nich nastáva neprípustnosť.

Ďalším známym výsledkom duality je nasledovná veta, známa tiež ako silná veta o dualite.

Veta 1.3.5. (Silná veta o dualite)

Ak má primárna úloha optimálne riešenie $\hat{x}$, tak aj duálna úloha má optimálne riešenie $\hat{y}$ a hodnoty účelových funkcií v optimách sa rovnajú $c^T \hat{x} = b^T \hat{y}$.

Zo silnej vety o dualite vyplýva, že ak jedna z dvojice vzájomne duálnych úloh (P), (D) má optimálne riešenie, tak aj druhá úloha má optimum.

Na základe predchádzajúcich tvrdení konštatujeme, že pre dvojicu vzájomne duálnych úloh (P) a (D) nastáva práve jedna z možností: obidve úlohy majú optimálne riešenie, jedna úloha je neohraničená a druhá vykazuje neprípustnosť alebo obidve úlohy sú neprípustné.

Nulovú veľkosť duálnej medzery právom možno považovať za postačujúcu a zároveň aj nutnú podmienku optimality. Navyše z podmienok primárno-duálnej prípustnosti vyplýva: $c^T x - b^T y = x^T c - (x^T A^T) y = x^T (c - A^T y) = x^T z$. Vidíme, že duálnu medzeru vyjadruje aj číslo $x^T z$.

Veta 1.3.6. (Veta o komplementarite)

Nech $x \in K$ je prípustné riešenie primárnej úlohy (P) a $(y, z) \in L$ prípustné riešenie duálnej úlohy (D), kde vektor z je vektor príslušných duálnych doplnkových premenných. Potom x a (y, z) sú optimálne riešenia pre primárnu (P) a duálnu (D) úlohu vtedy a len vtedy, ak platí tzv. podmienka komplementarity: $x^T z = 0_n$, t.j. $x_j z_j = 0, \quad j = 1, \ldots, n$.

Veta 1.3.7. (Veta o ostrej komplementarite)

Ak úlohy lineárneho programovania majú optimálne riešenia, potom existujú také optimálne riešenia $\hat{x}, (\hat{y}, \hat{z})$ primárnej, resp. duálnej úlohy, ktoré sú ostro komplementárne, t.j. $\hat{x} + \hat{z} > 0_n$.

Dôsledok 1.3.8 (Karush-Kuhn-Tuckerove podmienky optimality)

Nech $\hat{x} \in R^n, \hat{y} \in R^m, \hat{z} \in R^n$. *Potom* $\hat{x}$ *je optimálne riešenie úlohy* (P) *a* $(\hat{y}, \hat{z})$ *je optimálne riešenie úlohy* (D) *práve vtedy, keď platí:*

$$\begin{array}{rcccl} A\hat{x} & & = & b, & \hat{x} \geq 0_n \\ A^T\hat{y} & + \quad \hat{z} & = & c, & \hat{z} \geq 0_n . \\ \hat{x}^T\hat{z} & & = & 0 & \end{array} \tag{1.7}$$

Prvý riadok systému (1.7) reprezentuje primárnu prípustnosť, druhý riadok duálnu prípustnosť a posledná rovnosť podmienku komplementarity.

2. Metódy vnútorného bodu

2.1. Úvod do metód vnútorného bodu

Množina prípustných riešení úlohy (P) je polyedrická množina. Keďže účelová funkcia je lineárna, tak v prípade existencie optimálneho riešenia, toto riešenie musí ležať v niektorom z vrcholov danej polyedrickej množiny. Na nájdenie optimálneho riešenia úlohy (P) možno použiť simplexovú metódu. Idea simplexového algoritmu je hľadať optimálne riešenie vo vrcholoch polyedrickej množiny pomocou iteračného presunu po hranách tejto množiny tak, aby účelová funkcia zaznamenala pokles hodnoty. Pokiaľ už z daného vrcholu nie je možný takýto posun do iného, tak simplexový algoritmus skončí a aktuálny vrchol vyhlási za optimálne riešenie.

Naopak, algoritmy metód vnútorného bodu štartujú z vnútra množiny prípustných riešení a optimálne riešenie hľadajú ako limitu postupnosti optimálnych riešení parametrických transformovaných úloh, odvodených z pôvodných úloh (P) a (D). Tieto metódy generujú nekonečnú postupnosť vnútorných bodov a algoritmus skončí, keď nájde bod dostatočne blízko optimálneho riešenia, ležiaceho na hranici množiny prípustných riešení. Toto je jeden z hlavných znakov metód vnútorného bodu, ktorým sa odlišujú od simplexovej metódy, ktorá po konečnom počte krokov dosiahne exaktné optimálne riešenie.

Metódy vnútorného bodu konvergujú k optimálnemu riešeniu, no skončia „iba“ v jeho dostatočne malom okolí. Avšak tento zjavný nedostatok nemá z hľadiska numeriky rušivý dopad, pretože väčšinou približné riešenie (s presnosťou rádovo $10^{-6}, 10^{-8}, \dots$) je postačujúce, resp. procesy zaokrúhľovania sú schopné „skoro“ optimálny vnútorný bod konvertovať na presné vrcholové optimálne riešenie.

Medzi ďalšie rozdiely medzi metódami vnútorného bodu a simplexovou metódou patrí výpočtová zložitosť. Simplexový algoritmus môže mať v najhoršom prípade exponenciálnu zložitosť. Metódy vnútorného bodu potrebujú na dosiahnutie požadovanej presnosti najviac polynomiálne mnoho operácií.

2.2. Rozdelenie metód vnútorného bodu

Je dôležité uvedomiť si, že v súbore metód vnútorného bodu existuje veľa rozličných algoritmov, ktoré môžu zdieľať niektoré spoločné základné princípy, avšak sa odlišujú svojimi individuálnymi charakteristikami. Medzi najčastejšie používané kritéria na klasifikáciu metód vnútorného bodu patria: priestor iterácií, typ iterovania, typ kroku a typ algoritmu.

Metódy delíme na primárne, duálne a primárno-duálne na základe toho, v ktorom priestore metódy iterujú, teda či jednotlivé iterácie patria do primárneho priestoru, duálneho priestoru alebo do kartézskeho súčinu týchto dvoch priestorov. V súčasnosti najčastejšie používané metódy sú metódy tretieho typu, ktoré súčasne riešia primárnu aj duálnu úlohu.

Metódy nazývame prípustné, ak ich iterácie spĺňajú podmienky prípustnosti, čiže vyhovujú ohraničeniam v tvare rovníc, aj podmienkam nezápornosti. V prípade neprípustných metód nemusia pre iterácie platiť rovnice väzieb, avšak stále vyžadujeme splnenie ich nezápornosti.

Niektoré algoritmy sú nútené robiť v každej iterácii veľmi malé kroky, čo má za následok enormné zvýšenie celkového počtu iterácií. Tieto metódy voláme metódy s krátkym krokom a zväčša sú odsunuté len do teoretického záujmu. Metódy viac používané v praxi sú tie, ktoré dovoľujú robiť väčšie kroky, odkiaľ majú odvodený názov metódy s dlhým krokom. Špeciálne možno voliť krok jednotkovej alebo optimálnej dĺžky.

Najväčšia rôznorodosť nastáva pri delení metód podľa typu algoritmu. Hoci ešte stále nie sú úplne štandardizované pomenovania jednotlivých typov algoritmu, v hrubom ponímaní možno metódy rozdeliť na metódy sledujúce centrálnu trajektóriu („path-following methods“), afinne-škálovacie metódy („affine-scaling methods“) a redukcie-schopné metódy(„potential reduction methods“).

2.3. Bariérová funkcia

Metódy vnútorného bodu patria do širokej triedy transformačných metód konvexného programovania. Špeciálne medzi parametrické bariérové metódy s lineárnym parametrom.

Odteraz budeme používať nasledovné značenie zaužívané v problematike metód vnútorného bodu. Definujme jednotkový vektor $e = (1,\ldots,1)^T$. Malými písmenami (napr. x) budeme, ako zvyčajne, označovať stĺpcové vektory $x = (x_1,\ldots,x_n)^T$. Prislúchajúcimi veľkými písmenami (napr. X) definujeme diagonálne matice, ktorých diagonálne prvky sú prvky príslušného vektora

$$X = diag(x) = \begin{pmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & 0 & x_n \end{pmatrix}.$$

Majme primárnu a k nej prislúchajúcu duálnu úlohu v štandardnom tvare

$$(P) \quad Min\left\{ c^T x \,\middle|\, Ax = b, x \geq 0_n \right\},$$

$$(D) \quad Max\left\{ b^T y \,\middle|\, A^T y + z = c, z \geq 0_n \right\}.$$

Bod x nazývame ostro prípustným primárnym bodom, ak $x > 0$ a $x \in K$. Množinu všetkých ostro prípustných primárnych bodov označujeme

$$K^0 = \left\{ x \in R^n \,\middle|\, Ax = b, x > 0_n \right\}. \tag{2.1}$$

Podobne môžeme zadefinovať množinu všetkých ostro prípustných duálnych bodov

$$L^0 = \left\{ (y,z) \in R^m \times R^n \,\middle|\, A^T y + z = c, z > 0_n \right\}. \tag{2.2}$$

V algoritmoch metód vnútorného bodu sú dôležité nasledovné dva predpoklady, ktorých splnenie budeme odteraz predpokladať.

Predpoklad 2.3.1.

Matica A má plnú hodnosť, t.j. $h(A) = m$, pričom $m < n$.

Tento predpoklad nám zabezpečuje jedno-jednoznačný vzťah dvojice duálnych premenných v množine duálne prípustných riešení. Metódy vnútorného bodu hľadajú optimálne riešenie úloh (P) a (D) z vnútra množín prípustných riešení K a L (čiže na množinách K^0 a L^0) ako limitu postupnosti riešení úloh odvodených z (P) a (D). Preto je dôležitý aj druhý predpoklad.

Predpoklad 2.3.2.

Množiny K^0 a L^0 sú neprázdne.

Vráťme sa k primárnej úlohe

$$(P) \qquad Min\left\{ c^T x \,\middle|\, Ax = b, x \geq 0_n \right\}.$$

Prítomnosť neostrých nerovností vyskytujúcich sa v podmienkach nezápornosti premenných predstavuje problematickosť výpočtu optimálneho riešenia. Odstrániť „problematické =" v podmienke $x \geq 0_n$ sa podarí tak, že jednotlivé zložky vektora x_j, $j = 1, \dots n$ budú argumentmi tzv. aditívneho bariérového členu v novej účelovej funkcii. Namiesto pôvodnej účelovej funkcie $c^T x$ budeme na množine K^0 minimalizovať parametrickú bariérovú funkciu, ktorá automaticky zabezpečí platnosť podmienky $x \geq 0_n$. Nová účelová funkcia je

$$T(x,\mu) = c^T x + \mu \sum_{j=1}^{n} \Gamma(x_j), \tag{2.3}$$

kde $\mu > 0$ je riadiaci parameter.

Bariérová transformácia by mala spĺňať niekoľko predpokladov. Γ je dva krát spojite diferencovateľná funkcia $\Gamma : R_{++} \to R$, $\xi \in R_{++}$ a platí: konvexnosť (C), monotónnosť (M), bariérovosť (B), resp. kvázibariérovosť (QB), asymptotickosť (A):

- (C) $\Gamma(\xi)$ je konvexná ($\Gamma'(\xi)$ je rastúca),
- (M) $\Gamma(\xi)$ je monotónna (klesajúca),

- (B) $\lim_{\xi\downarrow 0}\Gamma(\xi)=+\infty \quad \left(\Rightarrow \lim_{\xi\downarrow 0}\Gamma'(\xi)=-\infty\right)$,
- (QB) $\lim_{\xi\downarrow 0}\Gamma(\xi)=\Gamma(0)$ a $\lim_{\xi\downarrow 0}\Gamma'(\xi)=-\infty$,
- (A) $\lim_{\xi\to+\infty}\Gamma'(\xi)=0$.

Typickými príkladmi bariérových transformácií sú funkcie

$$\Gamma_1(\xi)=-\ln(\xi), \tag{2.4}$$

$$\Gamma_2(\xi)=\frac{1}{\xi} \tag{2.5}$$

a kvázibariérová funkcia

$$\Gamma_3(\xi)=-\sqrt{\xi}\,. \tag{2.6}$$

Riešenie pôvodnej úlohy (P) sa nám pretransformuje na riešenie modifikovanej parametrickej úlohy na voľný extrém

$$(P_\mu) \quad Min\{T(x,\mu)|\ Ax=b\} \tag{2.7}$$

s parametrom $\mu>0$. Pre fixovanú hodnotu parametra $\mu>0$ je bariérová funkcia $T(x,\mu)$ rýdzo konvexná na množine K^0, preto má vo vnútri množiny primárne prípustných riešení nanajvýš jeden bod minima. Ak je množina $\hat{K}$ optimálnych riešení úlohy (P) neprázdna a ohraničená, tak $T(x,\mu)$ má minimum vždy.

2.4. Centrálna trajektória

Modifikovaná úloha (P_μ) v skutočnosti nie je ekvivalentná s úlohou (P), ale je to celá trieda úloh závislá od parametra $\mu>0$. Ak μ je blízke nule, potom sa úloha (P_μ) „blíži“ k úlohe (P) v zmysle, že riešenia úloh (P_μ) konvergujú k riešeniu (P) pre $\mu\downarrow 0$. Polyedrická množina K prípustných riešení má v ľubovoľnom bode steny aspoň jednu premennú nulovú. Preto účelová funkcia $T(x,\mu)$ pri fixnom μ dosahuje na hranici množiny K nekonečne veľkú hodnotu. Tak vytvára „bariéru“, ktorá odtláča minimalizujúcu postupnosť vnútorných bodov od hranice. Minimum tak dosahuje funkcia $T(x,\mu)$ vo vnútri množiny K - na množine K^0. Ak sa na riešenia postupnosti úloh (P_μ) pozrieme ako na funkciu parametra μ, tak množina optimálnych riešení vytvára trajektóriu cez vnútro množiny P, ktorú nazývame primárna centrálna trajektória.

Pre duálnu úlohu

$$(D) \quad Max\left\{ b^T y \,\middle|\, A^T y + z = c,\, z \geq 0_n \right\}$$

analogickým spôsobom definujeme transformačnú bariérovú funkciu

$$Q(y,z,\mu) = b^T y - \mu \sum_{j=1}^{n} \Gamma(z_j) \tag{2.8}$$

s parametrom $\mu > 0$ a parametrickú triedu úloh

$$(D_\mu) \quad Max\left\{ Q(y,z,\mu) \,\middle|\, A^T y + z = c \right\}, \tag{2.9}$$

ktorých riešenia závislé od voľby parametra μ, tvoria duálnu centrálnu trajektóriu - krivku vo vnútri množiny L duálne prípustných riešení.

3. Vybrané metódy

Význam prechodu od úloh (P) a (D) k úlohám (P_μ) a (D_μ) spočíva v tom, že na riešenie modifikovaných úloh (P_μ) a (D_μ) pre fixné hodnoty parametra μ možno použiť Lagrangeovu metódu. Lagrangeove podmienky v tomto prípade predstavujú nutné aj postačujúce podmienky optimality riešenia $x(\mu)$ úlohy (P_μ) a riešenia $(y(\mu), z(\mu))$ úlohy (D_μ).

3.1. Logaritmická bariérová metóda

Pre obe z dvojice vzájomne duálnych úloh (P_μ) a (D_μ) dosaďme do bariérového členu funkcií (2.3) a (2.8) záporne vzatú logaritmickú funkciu $\Gamma^P(\xi) = \Gamma^P(\xi) = -\ln(\xi)$. Pôvodnú primárnu úlohu (P) nahradíme novou úlohou

$$(P_\mu) \quad Min\{T(x,\mu) \mid Ax = b\}, \tag{3.1}$$

$$T(x,\mu) = c^T x - \mu \sum_{j=1}^{n} \ln x_j\,,\ \mu > 0\,. \tag{3.2}$$

Lagrangeova funkcia pre úlohu (P_μ) je daná

$$L^P(x, y, \mu) = c^T x - \mu \sum_{j=1}^{n} \ln x_j - y^T(Ax - b),\ \mu > 0\,. \tag{3.3}$$

Jej prvé derivácie položené rovné nule nás privedú na nutné a postačujúce podmienky optimality pre $x > 0_n$ a $y \in R^m$

$$\nabla_y L^P = -Ax + b = 0_m\,, \tag{3.4}$$

$$\frac{\partial L^P}{\partial x_j} = c_j - \mu \frac{1}{x_j} - (A^T y)_j = 0, \quad j = 1,\ldots,n\,. \tag{3.5}$$

Pre rovnicu (3.5) zavedieme substitúciu

$$z_j = \mu \frac{1}{x_j} > 0, \quad j = 1,\ldots,n\,, \tag{3.6}$$

$$\text{t.j. } x_j z_j = \mu, \quad j = 1,\ldots,n \text{ a } z > 0_n\,. \tag{3.7}$$

Keďže požadujeme kladnosť vektorov $x > 0_n$ a $z > 0_n$, tak sa dá táto rovnosť ekvivalentne prepísať do foriem: $Xz = \mu e \iff XZe = \mu e$. A teda systém podmienok optimality nadobudne tvar:

$$\begin{array}{rcrcll} & & Ax & = & b\ , & x > 0_n \\ A^T y & + & z & = & c\ , & z > 0_n\ . \\ & & XZe & = & \mu e & \end{array} \tag{3.8}$$

Podobne pôvodnú duálnu úlohu (D) transformujeme na novú úlohu s novou účelovou funkciou:

$$(D_\mu) \quad Max\left\{Q(y,z,\mu) \middle| \ A^T y + z = c\right\}, \tag{3.9}$$

$$Q(y,z,\mu) = b^T y + \mu \sum_{j=1}^{n} \ln z_j\ ,\ \mu > 0\ . \tag{3.10}$$

Lagrangeova funkcia pre (D_μ) bude

$$L^D(y,z,x,\mu) = b^T y + \mu \sum_{j=1}^{n} \ln z_j - x^T\left(A^T y + z - c\right),\ \mu > 0 \tag{3.11}$$

a od nej odvodené nutné a postačujúce podmienky optimality pre $y \in R^m$ a $z > 0_n$ sú

$$\nabla_y L^D = b - Ax = 0_m, \tag{3.12}$$

$$\nabla_x L^D = -A^T y - z + c = 0_n, \tag{3.13}$$

$$\frac{\partial L^D}{\partial z_j} = \mu \frac{1}{z_j} - x_j = 0,\quad j = 1,\ldots,n\ . \tag{3.14}$$

Tretia rovnica (3.14) $z_j x_j = \mu,\ \ j = 1,\ldots,n$, pre $x > 0_n$ sa dá opäť upraviť, keďže žiadame kladnosť vektorov $x > 0_n$ a $z > 0_n$, na rovnosti $Zx = \mu e \iff ZXe = XZe = \mu e$. Pre duálnu úlohu dostávame rovnakú sústavu rovníc ako pre úlohu primárnu, t.j. systém (3.8) je spoločný systém pre (P_μ) a (D_μ). Je charakterizovný:

$$\begin{array}{rcrcll} & & Ax & = & b\ , & x > 0_n \ \ldots\ldots\ldots \text{primárna prípustnosť} \\ A^T y & + & z & = & c\ , & z > 0_n \ \ldots\ldots\ldots \text{duálna prípustnosť} \\ & & XZe & = & \mu e & \qquad\ \ \ldots\ldots\ldots\ \mu-\text{komplementarita} \end{array}$$

Lema 3.1.1.

Pre každé $\mu > 0$ majú úlohy (P_μ) a (D_μ) práve jedno riešenie.

Lema 3.1.2.*Pre každé* $\mu > 0$ *majú centrujúce podmienky (3.8) práve jedno riešenie* $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$.

Sústava modifikovaných Karush-Kuhn-Tuckerových (3.8) podmienok predstavuje nutné a postačujúce podmienky optimality pre úlohy (P_μ) a (D_μ). Nájdené riešenia $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$ tejto nelineárnej sústavy rovníc sa pre $\mu \downarrow 0$ limitne blížia k optimálnemu riešeniu $(\hat{x}, \hat{y}, \hat{z})$ úloh (P) a (D). Riešenia $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$ pre všetky hodnoty parametra $\mu > 0$ tvoria primárno-duálnu centrálnu trajektóriu. Definujeme ju teda ako krivku nasledovne:

$$\Lambda(\mu) = \left\{ (x(\mu), y(\mu), z(\mu)) : Ax = b, A^T y + z = c, XZe = \mu e, \mu > 0 \right\}. \qquad (3.15)$$

V teórii metód vnútorného bodu existujú algoritmy, ktoré generujú postupnosť bodov na tejto trajektórii, resp. blízko nej, a táto postupnosť pre $\mu \downarrow 0$ konverguje k ostro komplementárnemu optimálnemu riešeniu pôvodných úloh (P) a (D).

3.1.1. Newtonova metóda

Hľadanie optimálneho riešenia pôvodných úloh (P) a (D) je ekvivalentné riešeniu série modifikovaných úloh (P_μ) a (D_μ) pre hodnoty parametra $\mu \downarrow 0$. Pre fixné μ má každá úloha optimálne riešenie závislé od μ, ktoré musí spĺňať nutné a postačujúce podmienky (3.8). Úlohu riešiť tento systém môžeme jednoducho previesť na úlohu hľadania nulového bodu funkcie $F(x, y, z) : R^{2n+m} \to R^{2n+m}$, ktorá má pre logaritmickú bariérovú funkciu tvar

$$F(x, y, z) = \begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe - \mu e \end{pmatrix}.$$

Najpopulárnejšia metóda riešenia takéhoto nelineárneho systému je bezpochyby Newtonova metóda, ktorej základné princípy popíšeme v najbližších odstavcoch.

Nech $F : R^N \to R^N$ je diferencovateľné zobrazenie

$$F(\xi) = \begin{bmatrix} F_1(\xi) \\ F_2(\xi) \\ \vdots \\ F_N(\xi) \end{bmatrix}, \; \xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_N \end{bmatrix}.$$

Newtonova metóda je iteračná metóda, ktorá má za účel nájsť taký bod $\xi^* \in R^N$, ktorý je koreňom rovnice $F(\xi) = 0_N$. Pre každú iteráciu $\xi_k \in R^N$ je cieľom nájsť smer $\Delta\xi \in R^N$

spĺňajúci $F(\xi_k + \Delta\xi) = 0_N$. Funkciu F v bode $\xi_{k+1} = \xi_k + \Delta\xi$ aproximujeme Taylorovým rozvojom prvého rádu

$$F(\xi_{k+1} + \Delta\xi) \approx F(\xi_k) + J(\xi_k)\Delta\xi\,, \tag{3.16}$$

kde $J(\xi_k)$ je (predpokladáme regulárna) Jakobiho matica zobrazenia F v bode ξ_k

$$J(\xi_k) = \begin{bmatrix} \frac{\partial F_1}{\partial \xi_1}(\xi_k) & \frac{\partial F_1}{\partial \xi_2}(\xi_k) & \cdots & \frac{\partial F_1}{\partial \xi_N}(\xi_k) \\ \frac{\partial F_2}{\partial \xi_1}(\xi_k) & \frac{\partial F_2}{\partial \xi_2}(\xi_k) & \cdots & \frac{\partial F_2}{\partial \xi_N}(\xi_k) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_N}{\partial \xi_1}(\xi_k) & \frac{\partial F_N}{\partial \xi_2}(\xi_k) & \cdots & \frac{\partial F_N}{\partial \xi_N}(\xi_k) \end{bmatrix}.$$

Položme pravú stranu (3.16) rovnú nule a získame lineárny systém pre hľadané $\Delta\xi$

$$J(\xi_k)\Delta\xi = -F(\xi_k). \tag{3.17}$$

Na konci každej iterácie aktualizujeme riešenie

$$\xi_{k+1} = \xi_k + \Delta\xi \tag{3.18}$$

a proces pokračuje pre $k = k+1$, kým nemáme uspokojujúce riešenie $F(\xi^*) \approx 0_N$.

3.1.2. Newtonova metóda pre logaritmickú bariérovú funkciu

Ako sme už naznačili, sústavu (3.8) môžeme formálne zapísať $F(x, y, z) = 0_{2n+m}$, kde funkcia $F(x, y, z): R^{2n+m} \to R^{2n+m}$, pričom požadujeme $x > 0$, $z > 0$. Predpokladáme, že v k-tej iterácii máme k dispozícii aproximáciu (x_k, y_k, z_k) hľadaného riešenia sústavy rovníc (3.8), pričom je ostro prípustné, teda vyhovuje podmienkam primárnej a duálnej prípustnosti a $x_k > 0$, $z_k > 0$. Potom

$$-F(x_k, y_k, z_k) = -\begin{pmatrix} Ax_k - b \\ A^T y_k + z_k - c \\ X_k Z_k e - \mu e \end{pmatrix} = \begin{pmatrix} 0_m \\ 0_n \\ \mu e - X_k Z_k e \end{pmatrix}. \tag{3.19}$$

Jakobiho matica bude

$$J(x_k, y_k, z_k) = \begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z_k & 0 & X_k \end{bmatrix}$$

a Newtonova sústava nadobudne tvar

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z_k & 0 & X_k \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 0_m \\ 0_n \\ \mu e - X_k Z_k e \end{bmatrix}. \tag{3.20}$$

Riešenie sústavy $F(x, y, z) = 0_{2n+m}$ v $(k+1)$-vej iterácii $(x_{k+1}, y_{k+1}, z_{k+1})$ bude definované:

$$\begin{aligned} x_{k+1} &= x_k + \Delta x \\ y_{k+1} &= y_k + \Delta y, \\ z_{k+1} &= z_k + \Delta z \end{aligned} \tag{3.21}$$

kde krok volíme jednotkový a $(\Delta x, \Delta y, \Delta z)$ sú Newtonove smery, keďže sú riešením Newtonovej sústavy:

$$\begin{aligned} A(\Delta x) & & &= 0_m \\ & A^T(\Delta y) + (\Delta z) & &= 0_n \quad . \\ Z_k(\Delta x) & + X_k(\Delta z) & &= \mu e - X_k Z_k e \end{aligned} \tag{3.22}$$

Matica tejto sústavy je pomerne riedka, čo umožňuje elimináciou Δx, Δz ľahko vyjadriť hľadané zložky smerového vektora. Vektor Δy získame z menšej sústavy:

$$\left(A X_k Z_k^{-1} A^T\right) \Delta y = -A Z_k^{-1} (\mu e - X_k Z_k e) \tag{3.23}$$

a zvyšné vektory Δx, Δz jednoducho dopočítame zo vzťahov

$$\begin{aligned} \Delta z &= -A^T(\Delta y) \\ \Delta x &= Z_k^{-1}(\mu e - X_k Z_k e) - Z_k^{-1} X_k (\Delta z) \end{aligned}. \tag{3.24}$$

Prvé dve rovnice v (3.8) sú lineárne, a teda všetky nasledujúce newtonovské iterácie ich budú spĺňať. Avšak nemáme zaručenú požadovanú kladnosť zložiek $x_{k+1} > 0$, $z_{k+1} > 0$. Na zabezpečenie jej platnosti použijeme namiesto (3.21) modifikovanú iteračnú schému s korigovanou dĺžkou kroku:

$$\begin{aligned} x_{k+1} &= x_k + \hat{\alpha} \Delta x, \quad & \hat{\alpha} > 0 \\ y_{k+1} &= y_k + \hat{\beta} \Delta y, \quad & \hat{\beta} > 0 \\ z_{k+1} &= z_k + \hat{\beta} \Delta z & \end{aligned} \tag{3.25}$$

kde $\hat{\alpha} > 0$ volíme z intervalu $(0; \overline{\alpha})$, resp. $\hat{\beta} > 0$ z intervalu $(0; \overline{\beta})$, kde „kritické“ hranice dĺžky krokov $\overline{\alpha}, \overline{\beta}$ určujú maximálny prípustný krok v danom smere. Definujeme ich nasledovne:

$$\overline{\alpha} = \min\left\{ -\frac{(x_k)_j}{(\Delta x)_j} \,\middle|\, (\Delta x)_j < 0,\ j = 1,\dots,n \right\}, \tag{3.26}$$

$$\overline{\beta} = \min\left\{ -\frac{(z_k)_j}{(\Delta z)_j} \,\middle|\, (\Delta z)_j < 0,\ j = 1,\dots,n \right\}. \tag{3.27}$$

V prípade, že všetky zložky smerového vektora Δx sú nezáporné, tak z bodu x_k sa môžeme posunúť smerom Δx ľubovoľne ďaleko bez strachu, že by sme narazili na hranicu množiny K^0. Hodnotu $\hat{\alpha}$ optimálnej dĺžky kroku môžeme vtedy voliť ľubovoľne veľkú, a preto položíme $\overline{\alpha} = \infty$. Podobne postupujeme aj pri určovaní hodnoty $\overline{\beta}$ v prípade, keď zápornú súradnicu neobsahuje smerový vektor Δz.

Ľahko sa dá overiť, že podmienky prípustnosti pre nové riešenie ostanú zachované aj pri voľbe nejednotkovej dĺžky krokov. Optimálnu dĺžku krokov $\hat{\alpha} > 0$, $\hat{\beta} > 0$ zvolíme tak, aby nové riešenie minimalizovalo primárnu funkciu T, maximalizovalo duálnu funkciu Q a aby ležalo vo vnútri množiny prípustných riešení

$$\begin{aligned} \hat{\alpha} &= \arg\min \left\{ \varphi(\alpha) = T(x + \alpha(\Delta x), \mu) \,\middle|\, x + \alpha(\Delta x) \in K^0, \alpha \in (0; \overline{\alpha}) \right\}, \\ \hat{\beta} &= \arg\max \left\{ \psi(\beta) = Q((y,z) + \beta(\Delta y, \Delta z), \mu) \,\middle|\, (y,z) + \beta(\Delta y, \Delta z) \in L^0, \beta \in (0; \overline{\beta}) \right\}. \end{aligned} \tag{3.28}$$

Pokiaľ nová iterácia $(x_{k+1}, y_{k+1}, z_{k+1})$ aproximuje hľadané riešenie $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$ úloh (P_μ), (D_μ) dostatočne presne, iteračný Newtonov proces môžeme zastaviť. V opačnom prípade celú procedúru získavania novej iterácie opakujeme pre $k = k+1$. Mieru aproximácie riešenia $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$ bodom $(x_{k+1}, y_{k+1}, z_{k+1})$ predstavuje relatívna dĺžka rezídua v tretej rovnici systému (3.8). Kritérium zastavenia iteračného procesu teda môže mať tvar:

$$\frac{\| \mu e - X_{k+1} Z_{k+1} e \|}{\| \mu e \|} < \varepsilon_2, \tag{3.29}$$

kde $\varepsilon_2 > 0$ je zadaná presnosť.

3.2. Nelogaritmická bariérová metóda

Pre modifikované úlohy (P_μ), (D_μ) zvolíme zovšeobecnenú nelogaritmickú bariéru, avšak s rozličnými konštantami κ, ω pre primárnu a duálnu úlohu:

$$\Gamma^P(\xi) = -\frac{1}{\kappa}\xi^\kappa,\ \Gamma^D(\xi) = -\frac{1}{\omega}\xi^\omega,$$

ktoré spĺňajú podmienky $\kappa < 1$, $\kappa \neq 0$ a $(1-\kappa)(1-\omega) = 1$. Nové úlohy budú

$$\left(P_\mu\right) \quad Min\left\{T(x,\mu)\,\middle|\, Ax=b\right\}, \tag{3.30}$$

$$T(x,\mu)=c^T x-\frac{\mu}{\kappa}\sum_{j=1}^{n} x_j^\kappa\,,\ \mu>0\,,\ \kappa<1\,,\ \kappa\neq 0\,, \tag{3.31}$$

$$\left(D_\mu\right) \quad Max\left\{Q(y,z,\mu)\,\middle|\, A^T y+z=c\right\}, \tag{3.32}$$

$$Q(y,z,\mu)=b^T y+\frac{\mu^{1-\omega}}{\omega}\sum_{j=1}^{n} z_j^\omega\,,\ \mu>0\,,\ (1-\kappa)(1-\omega)=1\,. \tag{3.33}$$

Lagrangeove funkcie pre $\left(P_\mu\right)$ a $\left(D_\mu\right)$ sú

$$L^P(x,y,\mu)=c^T x-\frac{\mu}{\kappa}\sum_{j=1}^{n} x_j^\kappa-y^T(Ax-b),\ \mu>0\,,\ \kappa<1\,,\ \kappa\neq 0\,, \tag{3.34}$$

$$L_D(y,z,x,\mu)=b^T y+\frac{\mu^{1-\omega}}{\omega}\sum_{j=1}^{n} z_j^\omega-x^T\left(A^T y+z-c\right),\ \mu>0\,,\ (1-\kappa)(1-\omega)=1\,. \tag{3.35}$$

Rovnakým postupom ako pre logaritmickú metódu odvodíme nutné a postačujúce podmienky optimality. Pre úlohu $\left(P_\mu\right)$ máme $x>0_n$ a $y\in R^m$ a

$$\nabla_y\, L^P=-Ax+b=0_m \tag{3.36}$$

$$\frac{\partial L^P}{\partial x_j}=c_j-\mu\, x_j^{\kappa-1}-\left(A^T y\right)_j=0\,,\ j=1,\ldots,n\,. \tag{3.37}$$

Substitúciou $z_j=\mu\, x_j^{\kappa-1}>0$, $j=1,\ldots,n$, upravíme predchádzajúcu rovnosť (3.37) na tvary $x_j^{1-\kappa} z_j=\mu \Leftrightarrow X^{1-\kappa} z=\mu e \Leftrightarrow X^{1-\kappa} Ze=\mu e$, pričom stále chceme kladné vektory $x>0_n$ a $z>0_n$.

Pre úlohu $\left(D_\mu\right)$ a $y\in R^m$, $z>0_n$ máme

$$\nabla_y\, L^D=b-Ax=0_m\,, \tag{3.38}$$

$$\nabla_x\, L^D=-A^T y-z+c=0_n\,, \tag{3.39}$$

$$\frac{\partial L^D}{\partial z_j}=\mu^{1-\omega}\, z_j^{\omega-1}-x_j=0\,,\ j=1,\ldots,n\,, \tag{3.40}$$

kde tretiu rovnicu ekvivalentnú s rovnosťou $x_j=\mu^{1-\omega}\, z_j^{\omega-1}>0$, $j=1,\ldots,n$ prepíšeme do tvaru $z_j^{1-\omega} x_j=\mu^{1-\omega} \Leftrightarrow Z^{1-\omega}\, x=\mu^{1-\omega}\, e \Leftrightarrow Z^{1-\omega}\, Xe=\mu e$ a navyše $x>0_n$ a $z>0_n$.

Pre nelogaritmickú bariérovú funkciu dostávame dva systémy podmienok optimality – systém (3.41) pre úlohu (P_μ) a systém (3.42) pre úlohu (D_μ):

$$\begin{array}{rcll} Ax & = & b\ , & x > 0_n \\ A^T y + z & = & c\ , & z > 0_n \\ X^{1-\kappa} Ze & = & \mu e & \end{array} \tag{3.41}$$

$$\begin{array}{rcll} Ax & = & b\ , & x > 0 \\ A^T y + z & = & c\ , & z > 0\ , \\ Z^{1-\omega} Xe & = & \mu^{1-\omega} e & \end{array} \tag{3.42}$$

kde $\kappa < 1$, $\kappa \neq 0$ a platí $(1-\kappa)(1-\omega) = 1$.

3.2.1. Newtonova metóda pre nelogaritmickú bariérovú funkciu

Sústava (3.41), resp. (3.42) predstavuje nutné a postačujúce podmienky optimality pre úlohu (P_μ), resp. (D_μ). Umocnením poslednej rovnice v 2. systéme výrazom $(1-\kappa)$ sa druhá sústava dá previesť na rovnaký algebraický tvar ako prvá. Avšak ako ukážeme, Newtonova metóda aplikovaná na obe z nich vedie k dvom rôznym neekvivalentným sústavám. Považujeme ich teda za dve verzie - primárnu a duálnu verziu modifikovaných KKT podmienok optimality pre nelogaritmickú bariérovú metódu. Každá forma vedie k optimu $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$ „svojou cestou", a preto sú obidve kandidátmi na použitie pri výpočte novej iterácie. Odpoveď na otázku, ktorý z variantov smerov si vybrať na nájdenie nového bodu postupnosti riešení, súvisí s otázkou spádovosti smerov pre účelové funkcie T a Q, o čom hovorí nasledujúca veta:

Veta 3.2.1.

Nech $x_k \in K^0$ *a* $(y_k, z_k) \in L^0$. *Nech* $(\Delta x^P, \Delta y^P, \Delta z^P)$, $\Delta x^P \neq 0$ *je riešením Newtonovej sústavy (3.41) a* $(\Delta x^D, \Delta y^D, \Delta z^D)$, $\Delta z^D \neq 0$ *je riešením Newtonovej sústavy (3.42). Potom smer* Δx^P *je spádový pre funkciu* T *v bode* x_k *a smer* $(\Delta y^D, \Delta z^D)$ *je spádový pre funkciu* $-Q$ *v bode* (y_k, z_k).

Tieto výsledky sú dôvodom, prečo sa zaoberáme dvomi rozličnými postupmi hľadania riešenia $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$ úloh (P_μ), (D_μ). Zároveň dávajú návod na konštrukciu kombinovaného smeru pre Newtonove iterácie riešenia modifikovaných KKT podmienok.

Predpokladáme, že v k-tej iterácii máme ostro prípustný bod (x_k, y_k, z_k) aproximujúci riešenie $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$. Aplikujme na obe sústavy (3.41), (3.42) Newtonovu metódu. Primárne korekcie $(\Delta x^P, \Delta y^P, \Delta z^P)$ sú riešenia sústavy rovníc

$$\begin{array}{lcllcl} A(\Delta x) & & & & = & 0_m \\ & A^T(\Delta y) & + & (\Delta z) & = & 0_n \\ (1-\kappa) X_k^{-\kappa} Z_k (\Delta x) & & + & X_k^{1-\kappa}(\Delta z) & = & \mu e - X_k^{1-\kappa} Z_k e \end{array} . \qquad (3.43)$$

Newtonova metóda použitá na systém (3.42) nás privedie na sústavu rovníc (3.44) neekvivalentnú so sústavou (3.43), ktorej riešením bude duálna verzia smerov $(\Delta x^D, \Delta y^D, \Delta z^D)$

$$\begin{array}{lcllcl} A(\Delta x) & & & & = & 0_m \\ & A^T(\Delta y) & + & (\Delta z) & = & 0_n \\ Z_k^{1-\omega}(\Delta x) & & + & (1-\omega) X_k Z_k^{1-\omega}(\Delta z) & = & \mu^{1-\omega} e - Z_k^{1-\omega} X_k e \end{array} . \qquad (3.44)$$

Pre výpočet novej iterácie použijeme kombinovaný smer $(\Delta x, \Delta y, \Delta z) = (\Delta x^P, \Delta y^D, \Delta z^D)$. Dĺžku krokov opäť upravíme koeficientmi $\hat{\alpha} > 0$, $\hat{\beta} > 0$ a vkročíme do novej iterácie.

Riešenie našich dvoch Newtonových sústav je kľúčovým krokom algoritmu riešenia modifikovaných KKT podmienok pre nelogaritmickú bariérovú funkciu. Preto sa pokúsime zjednodušiť spôsob výpočtu smerov.

Prenásobením tretej rovnice (3.41) maticou X_k^κ zľava a tretej rovnice (3.42) maticou $(1-\kappa) Z_k^\omega$ zľava možno obe sústavy previesť na podobné sústavy rovníc. Sústavy majú rovnaké Jakobiho matice, avšak odlišujú sa jedným členom pravej strany. Ide o sústavu typu:

$$\begin{array}{lcllcl} A(\Delta x) & & & & = & 0_m \\ & A^T(\Delta y) & + & (\Delta z) & = & 0_n \\ (1-\kappa) Z_k (\Delta x) & & + & X_k(\Delta z) & = & r \end{array} , \qquad (3.45)$$

kde primárna pravá strana tretej rovnice je daná

$$r = \mu X_k^\kappa e - X_k Z_k e \qquad (3.46)$$

a duálna

$$r = (1-\kappa)\left(\mu^{1-\omega} Z_k^\omega e - Z_k X_k e\right). \qquad (3.47)$$

Rovnako ako pri logaritmickej transformácii je matica tohto systému riedka. Vektory smerov preto možno vypočítať zo sústavy:

$$\left(A X_k Z_k^{-1} A^T\right) \Delta y = -A Z_k^{-1} r \qquad (3.48)$$

a následne z vyjadrení:

$$\begin{aligned} \Delta z &= -A^T(\Delta y) \\ \Delta x &= (1-\omega) Z_k^{-1}\left(r - X_k(\Delta z)\right), \end{aligned} \tag{3.49}$$

pre r dané vzťahmi (3.46) a (3.47).

Použitím vypočítaných Newtonových smerov môžeme vkročiť do novej iterácie, ktorá je daná posunom zo starého bodu o optimálnu dĺžku kroku v smere daných korekcií

$$(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k, y_k, z_k) + (\Delta x, \Delta y, \Delta z).$$

Presnosť priblíženia k riešeniu $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$ bodom (x_k, y_k, z_k) predstavuje relatívna dĺžka rezíduí tretích rovníc oboch systémov (3.41), (3.42). Podmienka zastavenia kombinovaných Newtonových iterácií môže vyzerať nasledovne:

$$\left(\frac{\| \mu e - X_k^{1-\kappa} Z_k e \|}{\| \mu e \|} < \varepsilon_2 \right) \wedge \left(\frac{\| \mu e - Z_k^{1-\omega} X_k e \|}{\| \mu e \|} < \varepsilon_2 \right), \tag{3.50}$$

kde $\varepsilon_2 > 0$ je konštanta. Ak nie sú splnené podmienky ukončenia iteračného procesu, tak celý postup hľadania riešenia Newtonovho systému zopakujeme pre $k = k+1$.

3.3. Metóda s novou triedou hľadania smerov

V tejto kapitole sa budeme zaoberať novou metódou hľadania smerových vektorov. Uvažujme štandardné úlohy lineárneho programovania (P) a (D). Nájdenie optimálneho primárno-duálneho riešenia je ekvivalentné riešeniu μ-centrujúcich sústav rovníc s parametrom $\mu > 0$ pre $\mu \downarrow 0$

$$\begin{array}{rcccl} Ax & & & = & b\ ,\quad x > 0 \\ A^T y & + & z & = & c\ ,\quad z > 0\,. \\ & & XZe & = & \mu e \end{array} \tag{3.51}$$

Pre fixné $\mu > 0$ má sústava práve jedno optimálne riešenie $(\hat{x}(\mu), \hat{y}(\mu), \hat{z}(\mu))$. Postupnosť týchto bodov pre $\mu \downarrow 0$ tvorí centrálnu trajektóriu.

Zavedieme nové označenie. Nech $u, v \in R^n$, navyše $v_j \neq 0,\ \ j = 1, \dots, n$. Potom pod výrazom

$$\frac{u}{v} = \left(\frac{u_1}{v_1},\ \frac{u_2}{v_2},\ \dots,\ \frac{u_n}{v_n} \right)^T$$

budeme rozumieť vektor, ktorého zložky sú podiely zložiek daných vektorov. Zodpovedajúca diagonálna matica bude

$$diag\left(\frac{u}{v}\right) = \begin{pmatrix} \frac{u_1}{v_1} & 0 & \cdots & 0 \\ 0 & \frac{u_2}{v_2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \frac{u_n}{v_n} \end{pmatrix}.$$

Definujme funkciu $\phi \in C^1$, $\phi : R^n_{++} \to R^n_{++}$ a predpokladajme, že k nej existuje inverzná funkcia ϕ^{-1}. Systém (3.51) možno prepísať na ekvivalentný tvar

$$\begin{array}{rcrcrcl} Ax & & & = & b & , & x > 0 \\ A^T y & + & z & = & c & , & z > 0, \\ & \phi(XZe) & & = & \phi(\mu e) & & \end{array} \tag{3.52}$$

alternatívne:

$$\begin{array}{rcrcrcl} Ax & & & = & b & , & x > 0 \\ A^T y & + & z & = & c & , & z > 0. \\ & \phi\left(\frac{XZe}{\mu}\right) & & = & \phi(e) & & \end{array} \tag{3.53}$$

3.3.1. Newtonova metóda pre metódu s novou triedou hľadania smerov

Na druhý uvedený variant aplikujeme Newtonovu metódu. Výhoda tejto metódy spočíva v zadefinovaní nového vektora

$$\upsilon \in R^n, \ \upsilon = \sqrt{\frac{XZe}{\mu}},$$

$$\text{t.j. } \upsilon = \left(\sqrt{\frac{x_1 z_1}{\mu}}, \ \sqrt{\frac{x_2 z_2}{\mu}}, \ \cdots, \ \sqrt{\frac{x_n z_n}{\mu}}\right)^T,$$

ktorý je možné použiť na škálovanie lineárneho Newtonovho systému. Predpokladajme, že v k-tej iterácii máme ostro prípustné riešenie (x_k, y_k, z_k). Newtonov systém bude vyzerať nasledovne

$$\begin{array}{lcccl} A(\Delta x) & & & = & 0_m \\ & A^T(\Delta y) & + \quad (\Delta z) & = & 0_n \\ \frac{z}{\mu}\phi'\left(\frac{XZe}{\mu}\right)(\Delta x) & & + \quad \frac{x}{\mu}\phi'\left(\frac{XZe}{\mu}\right)(\Delta z) & = & \phi(e)-\phi\left(\frac{XZe}{\mu}\right) \end{array}. \quad (3.54)$$

Použitím vektora υ v ňom prepíšeme poslednú rovnicu do tvaru

$$\frac{z}{\mu}\phi'(\upsilon^2)(\Delta x) \quad + \quad \frac{x}{\mu}\phi'(\upsilon^2)(\Delta z) \quad = \quad \phi(e)-\phi(\upsilon^2). \quad (3.55)$$

Označme

$$d_x = \frac{\upsilon(\Delta x)}{x}, \; d_z = \frac{\upsilon(\Delta z)}{z}. \quad (3.56), (3.57)$$

Výsledná sústava so smermi $(d_x, \Delta y, d_z)$ bude daná:

$$\begin{array}{lcccl} \tilde{A}_x d_x & & & = & 0_m \\ & \tilde{A}_z^T(\Delta y) & + \; d_z & = & 0_n \\ d_x & & + \; d_z & = & q \end{array}, \quad (3.58)$$

kde sme zaviedli nové matice

$$\tilde{A}_x = A\, diag\left(\frac{x}{\upsilon}\right), \; \tilde{A}_z = A\, diag\left(\frac{\upsilon}{z}\right) \quad (3.59), (3.60)$$

a označili pravú stranu

$$q = \frac{\phi(e)-\phi(\upsilon^2)}{\upsilon\,\phi'(\upsilon^2)}. \quad (3.61)$$

Rôznou voľbou predpisu funkcie ϕ možno dospieť k rozdielnym výrazom q na pravej strane Newtonovej sústavy (3.58) a tým k odlišným výpočtom hľadaných smerov. Spomedzi všetkých metód patriacich do novej triedy určovania smerov uvedieme zopár príkladov. K lineárnej funkcii $\phi(\xi)=\xi$ prislúcha pravá strana $q=\upsilon^{-1}-\upsilon$, ku kvadratickej funkcii $\phi(\xi)=\xi^2$ výraz $q=\frac{1}{2}(\upsilon^{-3}-\upsilon)$ a všeobecne pre parameter $h>1$ a funkciu $\phi(\xi)=\xi^{\frac{h+1}{2}}$ možno docieliť $q=\frac{2}{h+1}(\upsilon^{-h}-\upsilon)$. Tu možno upozorniť na prepojenie medzi metódami s novou triedou hľadania smerov a metódami typu prediktor-korektor. Takto zadefinované pravé strany možno rozložiť na dve časti. Prvá zložka typu υ^{-h} v skutočnosti reprezentuje korektor smer a druhá $-\upsilon$ sa dá chápať ako prediktor smer. Prediktor má za účel dosiahnuť zmenšovanie duálnej medzery v každej iterácii, pokým korektor má za cieľ centrovanie riešení smerom k centrálnej trajektórii a udržiavanie iterácií v jej blízkosti.

Dosaďme do (3.58) špeciálne funkciu $\phi(\xi)=\sqrt{\xi}$, pre ktorú odvodíme člen na pravej strane $q=2(e-\upsilon)$. Následne riešenie sústavy (3.58) možno vyjadriť ako

$$\begin{array}{rcl} \Delta y & = & -\left(\tilde{A}_x\,\tilde{A}_z^T\right)^{-1}\tilde{A}_x\,q \\ d_z & = & -\tilde{A}_z^T\,(\Delta y) \\ d_x & = & q-(d_z) \end{array} \tag{3.62}$$

a po dosadení do pôvodných smerov sú zostávajúce dve hľadané korekcie dané vzorcami:

$$\Delta x = d_x\,\frac{x}{\upsilon},$$

$$\Delta z = d_z\,\frac{z}{\upsilon}.$$

Nový bod postupnosti riešení približujúcich sa k optimálnemu riešeniu $(\hat{x},\hat{y},\hat{z})$ vznikne krokom jednotkovej dĺžky zo starého bodu smerom $(\Delta x,\Delta y,\Delta z)$. Symbolicky:

$$(x_{k+1},y_{k+1},z_{k+1})=(x_k,y_k,z_k)+(\Delta x,\Delta y,\Delta z).$$

Lema 3.3.1.

Nech $\delta=\delta(XZe,\mu)<1$ *a* $x_{k+1}=x_k+\Delta x$, $z_{k+1}=z_k+\Delta z$. *Potom* $x_{k+1}>0$ *a* $z_{k+1}>0$*, čiže plný Newtonov krok je ostro prípustný.*

Algoritmus začínajúci v ľubovoľnom vnútornom bode sa po niekoľko prvých iteráciach dostane do blízkosti centrálnej trajektórie a korektor časť kroku ju ustráži až do konca procesu. Mieru priblíženia sa nového riešenia k bodu na centrálnej trajektórii pre každé fixné $\mu>0$ môžeme sledovať pomocou mierky

$$\delta(XZe,\mu)=\frac{\|q\|}{2}=\|e-\upsilon\|=\left\|e-\sqrt{\frac{XZe}{\mu}}\right\|. \tag{3.63}$$

4. Numerické experimenty

V tejto kapitole sa budeme zaoberať praktickým použitím všetkých troch metód podrobne popísaných v predchádzajúcej kapitole. V prvej fáze experimentov sa pokúsime pre každú z metód nájsť najvhodnejšiu voľbu parametrov. Druhá fáza nám dá odpoveď, ktorá z metód vykazuje v praxi najlepšie výsledky.

Program, v ktorom prebiehajú všetky výpočty, bol vytvorený v programovacom prostredí MatLab 6.5. Výpis jeho zdrojového kódu a výsledky experimentu sú obsahom príloh.

4.1. Popis programu

Aby sme mohli uskutočniť vytýčený numerický experiment, zostavili sme generátor úloh, ktorý automaticky vygeneruje úlohy (P) a (D) spolu s ich ostro prípustnými optimálnymi a štartovacími riešeniami. Vstupom pre generátor sú požadované rozmery úloh m a n ($m < n$). Výstupom generátora sú matica A a vektory c, b, x_0, y_0, z_0, $\hat{x}$, $\hat{y}$, $\hat{z}$ príslušných rozmerov a optimálna hodnota účelových funkcií $\hat{\chi}$.

Na základe skutočnosti, že z výstupov generátora máme k dispozícii aj vygenerované optimálne riešenie $(\hat{x}, \hat{y}, \hat{z})$ a optimálnu hodnotu účelových funkcií $\hat{\chi} = c^T \hat{x} - b^T \hat{y}$, budeme pozorovať aj nasledovné odchýlky:

- duálna medzera: $c^T x - b^T y$ ... odchýlka hodnôt účelových funkcií v aktuálnom bode (x, y, z),
- primárna odchýlka: $c^T x - \hat{\chi}$ ... odchýlka hodnoty účelovej funkcie $c^T x$ v aktuálnom bode x od optimálnej hodnoty účelovej funkcie,
- duálna odchýlka: $\hat{\chi} - b^T y$ ... odchýlka hodnoty účelovej funkcie $b^T y$ v aktuálnom bode y od optimálnej hodnoty účelovej funkcie,
- vzdialenosti aktuálneho bodu od optimálneho riešenia: $\| x - \hat{x} \|$, $\| y - \hat{y} \|$, $\| z - \hat{z} \|$.

V prvej fáze experimentu vygenerujeme sadu piatich príkladov s rozmermi 80×100, ktoré sa pokúsime vyriešiť postupne všetkými tromi metódami, bližšie popísanými v tretej kapitole. Na základe výsledkov pre každú metódu vyberieme podľa nás najvhodnejšie hodnoty parametrov. V druhej fáze na vzorke 20 vytvorených úloh (tiež s rozmermi 80×100) porovnáme medzi sebou všetky tri metódy s konkrétnou voľbou parametrov zodpovedajúcich výsledkom z prvej fázy.

4.2. Klasické metódy

Zjednodušená idea primárno-duálnych metód vnútorného bodu spočíva v nasledovnom. Pôvodnú dvojicu vzájomne duálnych úloh (P) a (D) nahradíme triedou parametrických úloh (P_μ) a (D_μ). Pre dané $\mu > 0$ nájdeme približné primárno-duálne riešenie. Zmenšíme hodnotu parametra μ a postup zopakujeme. Štartovacím bodom pre túto minimalizáciu je vypočítané približné riešenie predchádzajúcej úlohy. Opakované riešenie týchto úloh pre hodnoty parametra $\mu \downarrow 0$ nazývame vonkajšie iterácie. Procesu hľadania postupnosti bodov približujúcich sa k optimálnemu riešeniu jednotlivých úloh (P_μ) a (D_μ) pre fixné $\mu > 0$ hovoríme vnútorná iterácia. Je to vlastne približovanie sa k bodu ležiacemu na centrálnej trajektórii. Na hľadanie tejto aproximácie použijeme modifikovanú Newtonovu metódu popísanú v kapitole 3.1.2 a 3.2.1. Na reguláciu dĺžky kroku použijeme algoritmus metódy zlatého rezu, pomocou ktorého získame „optimálny" krok.

Úlohou hľadania ostro prípustného štartovacieho bodu sa nemusíme zaoberať, keďže ho máme na začiatku k dispozícii z výstupov generátora.

Zastavovacie kritéria pre vnútorné Newtonove iterácie sme uviedli v kapitolách 3.1.2 a 3.2.1. Ako podmienku konca behu vonkajších iterácií možno zvoliť napríklad relatívnu veľkosť duálnej medzery

$$\frac{\| c^T x - b^T y \|}{1 + \| c^T x \|} < \varepsilon_1 \tag{4.1}$$

pre požadovanú presnosť $\varepsilon_1 > 0$.

Možnosti voľby rôznych hodnôt parametrov ε_1, ε_2, θ nám môžu ponúknuť rôzne presnosti výpočtov.

4.2.1. Algoritmy a výsledky

Schematický algoritmus prípustnej primárno-duálnej logaritmickej metódy vnútorného bodu:

<u>VSTUPY:</u> A, c, b, x_0, y_0, z_0

<u>PARAMETRE:</u> $\varepsilon_1 > 0$, $\varepsilon_2 > 0$ ... konštanty presnosti,

$\theta \in (0;1)$ ... parameter redukcie μ

COMPUTE $\mu_0 = \dfrac{x_0^T z_0}{n}$ ... bariérový parameter

SET $(x, y, z) = (x_0, y_0, z_0)$

$\mu = \mu_0$

WHILE $\dfrac{\| c^T x - b^T y \|}{1 + \| c^T x \|} > \varepsilon_1$ ((vonkajšia iterácia))

WHILE $\dfrac{\| \mu e - XZe \|}{\| \mu e \|} > \varepsilon_2$ ((vnútorná iterácia))

COMPUTE $(\Delta x, \Delta y, \Delta z)$=log_newton_smer

$(\hat{\alpha}, \hat{\beta})$=log_newton_krok

$(x, y, z) = (x, y, z) + (\hat{\alpha}\,\Delta x, \hat{\beta}\,\Delta y, \hat{\beta}\,\Delta z)$

END ((vnútorná iterácia))

$\mu = \theta\mu$

END ((vonkajšia iterácia))

Generovanú sadu piatich príkladov s rozmermi úloh 80×100 sme riešili logaritmickou bariérovou metódou pre zafixované hodnoty parametrov $\varepsilon_1 = 10^{-8}$, $\varepsilon_2 = 10^{-3}$ a skúmali sme vplyv zmeny parametra $\theta \in (0;1)$ na presnosť riešení. Všetky prislúchajúce tabuľky sú v prílohe.

PRIEMERY	ε_1 **= 1,E-08**	ε_2 **= 1,E-03**	
	θ **= 5,E-02**	θ **= 5,E-03**	θ **= 5,E-04**
μ_{END}	2,7103E-03	1,0841E-03	2,1682E-03
# vonkajších iterácií	6,00	4,00	3,00
# vnútorných iterácií	19,40	15,60	14,20
$c^T x - b^T y$	2,7106E-01	1,1209E-01	2,1716E-01
$c^T x - \hat{\chi}$	5,4242E-02	2,5373E-02	4,3721E-02
$\hat{\chi} - b^T y$	2,1682E-01	8,6718E-02	1,7344E-01
$\| x - \hat{x} \|$	1,1143E-02	4,4393E-03	8,8739E-03
$\| y - \hat{y} \|$	1,2783E-04	5,1283E-05	1,0234E-04
$\| z - \hat{z} \|$	7,4472E-03	2,9874E-03	5,9622E-03

Tabuľka 4.1 Priemerné hodnoty nameraných odchýlok riešení použitím logaritmickej metódy

Z hľadiska najnižšieho priemerného počtu vonkajších aj vnútorných iterácií sa núka ako najlepšia voľba parametra $\theta = 5.10^{-4}$. Avšak z pohľadu presnosti priemerných odchýlok pre jednotlivé voľby redukčného parametra sa zdá byť najvhodnejší výber hodnoty $\theta = 5.10^{-3}$, preto práve túto hodnotu zvolíme do druhej fázy.

Algoritmus nelogaritmickej bariérovej metódy je založený na približne rovnakých princípoch ako algoritmus pre logaritmickú funkciu. Hlavným rozdielom je asymetria medzi primárnou a duálnou zložkou problému. Naznačuje to aj prítomnosť ďalších parametrov – primárny parameter κ a duálny parameter ω, ktorých voľba ponúka ďalšie možnosti analýzy.

Vonkajšie iterácie procesu sú totožné ako pre logaritmickú funkciu. Asymetria sa prejavuje v potrebe riešiť v každej vnútornej iterácii Newtonovu sústavu pre primárne a pre duálne smery a vytvárať tak kombinovaný smer.

Schematický algoritmus prípustnej primárno-duálnej nelogaritmickej metódy vnútorného bodu:

<u>VSTUPY:</u> A, c, b, x_0, y_0, z_0

<u>PARAMETRE:</u> $\varepsilon_1 > 0$, $\varepsilon_2 > 0$ ... konštanty presnosti

$\theta \in (0;1)$ ... parameter redukcie μ,

$\kappa < 1, \kappa \neq 0$ ... parameter primárnej nelogaritmickej transformácie

COMPUTE $\mu_0 = \dfrac{x_0^T z_0}{n}$ ... bariérový parameter

$\omega = 1 - \dfrac{1}{1-\kappa}$ ... parameter duálnej nelogaritmickej transformácie

SET $(x, y, z) = (x_0, y_0, z_0)$

$\mu = \mu_0$

WHILE $\dfrac{\| c^T x - b^T y \|}{1 + \| c^T x \|} > \varepsilon_1$ ((vonkajšia iterácia))

WHILE $\left(\dfrac{\| \mu e - X_k^{1-\kappa} Z_k e \|}{\| \mu e \|} > \varepsilon_2 \right) \vee \left(\dfrac{\| \mu e - Z_k^{1-\omega} X_k e \|}{\| \mu e \|} > \varepsilon_2 \right)$ ((vnútorná iterácia))

COMPUTE $(\Delta x, \Delta y, \Delta z)$ = nelog_newton_smer

$(\hat{\alpha}, \hat{\beta})$ = nelog_newton_krok

$(x, y, z) = (x, y, z) + (\hat{\alpha}\,\Delta x, \hat{\beta}\,\Delta y, \hat{\beta}\,\Delta z)$

END ((vnútorná iterácia))

$\mu = \theta\mu$

END ((vonkajšia iterácia))

PRIEMERY ε_1 **= 1,E-08**	ε_2 **= 1,E-03** κ **= 1/3** ω **= - 1/2**	ε_2 **= 1,E-03** κ **= 1/10** ω **= - 1/9**	ε_2 **= 1,E-04** κ **= - 1/10** ω **= 1/11**	ε_2 **= 1,E-04** κ **= - 1/3** ω **= 1/4**
μ_{END}	1,4193E-03	4,9474E-03	1,5402E-04	6,1263E-04
# vonkajších iterácií	3,00	3,00	3,00	3,00
# vnútorných iterácií	32,60	15,40	21,60	17,80
$c^T x - b^T y$	8,1015E-01	7,4612E-01	3,6791E-02	2,9773E-01
$c^T x - \hat{\chi}$	2,8955E-04	3,1642E-02	2,9962E-02	2,9080E-01
$\hat{\chi} - b^T y$	8,0986E-01	7,1448E-01	6,8286E-03	6,9383E-03
$\lVert x - \hat{x} \rVert$	9,8675E-05	9,4614E-03	1,5527E-03	2,2755E-02
$\lVert y - \hat{y} \rVert$	1,4684E-04	2,9289E-04	5,8641E-06	1,4512E-05
$\lVert z - \hat{z} \rVert$	8,7748E-03	1,7185E-02	3,3943E-04	8,2925E-04

Tabuľka 4.2 Priemerné hodnoty nameraných odchýlok riešení použitím nelogaritmickej metódy

Pre všetky zvolené hodnoty parametrov $\kappa < 1, \kappa \neq 0$, θ vysvitla rovnaká potreba 3 vonkajších iterácií, no počty vnútorných cyklov sa rôznili. Najmenší priemerný počet Newtonových iterácií nastal pri voľbe parametrov $\kappa = 1/10, \theta = 10^{-3}$, no z hľadiska presnosti riešení a odchýlok funkčných hodnôt by to nebol dobrý výber. Za najuspokojivejšiu voľbu z použitých hodnôt parametrov sme do druhej fázy experimentov vybrali prípad $\kappa = -1/10, \theta = 10^{-4}$.

4.3. Nová metóda

Metóda s novou triedou hľadania smerov sa odlišuje od „klasických“ (logaritmických a nelogaritmických) metód. Hlavný rozdiel v algoritme je, že namiesto iterovania vo vonkajších a vnútorných cykloch a hľadania optimálnej dĺžky kroku táto metóda robí v každej iterácii plný Newtonov krok. Približovanie sa k optimálnemu riešeniu a sledovanie centrálnej trajektórie tu prebieha v každom kroku súčasne.

Schematický algoritmus prípustnej primárno-duálnej metódy vnútorného bodu s novou triedou hľadania smerov

VSTUPY: A, c, b, x_0, y_0, z_0

PARAMETRE: $\varepsilon > 0$, $\tau > 0$ ... konštanty presnosti

$\lambda = \dfrac{1}{2\sqrt{n}}$ ... parameter redukcie μ

COMPUTE $\mu_0 = \dfrac{x_0^T z_0}{n}$ ... bariérový parameter

$$\delta(X_0 Z_0 e, \mu_0) = \left\| e - \sqrt{\frac{X_0 Z_0 e}{\mu_0}} \right\|$$ ... blízkosť k centrálnej trajektórii

SET $(x, y, z) = (x_0, y_0, z_0)$

$\mu = \mu_0$

WHILE $x^T z > \varepsilon$

$\mu = (1-\lambda)\mu$

COMPUTE $(\Delta x, \Delta y, \Delta z)$=newclass_newton_smer

$(x, y, z) = (x, y, z) + (\Delta x, \Delta y, \Delta z)$

CHECK UP $\delta(XZe, \mu) < \tau$

END

PRÍKLAD 1	$\varepsilon = $ **1,E-02**	$\lambda = $ **5,E-02**	$\tau = $ **1/2**			
μ	9,6223E+03	9,1412E+03	8,6841E+03	8,2499E+03		9,6840E-05
δ	1,1191E+00	3,0741E-01	2,2582E-02	3,8155E-03		3,2879E-03
# iterácií	1	2	3	4	.....	360
$c^T x - b^T y$	8,8210E+05	9,0379E+05	8,6725E+05	8,2446E+05		9,6776E-03
$c^T x - \hat{\chi}$	1,1273E+05	1,1734E+05	1,1535E+05	1,1189E+05		9,3782E-03
$\hat{\chi} - b^T y$	7,6938E+05	7,8644E+05	7,5189E+05	7,1258E+05		7,7422E-03
$\\| x - \hat{x} \\|$	2,0725E+03	2,0918E+03	2,1022E+03	2,0911E+03		5,2745E-04
$\\| y - \hat{y} \\|$	7,5005E+00	7,0230E+00	6,8575E+00	6,5365E+00		3,0844E-06
$\\| z - \hat{z} \\|$	5,4020E+02	5,1490E+02	4,9380E+02	4,6623E+02		1,9523E-04

PRÍKLAD 2	$\varepsilon = $ **1,E-02**	$\lambda = $ **5,E-02**	$\tau = $ **1/2**			
μ	1,9463E+04	1,8490E+04	1,7566E+04	1,6687E+04		9,5526E-05
δ	1,0644E+00	3,9650E-01	5,2305E-02	3,7695E-03		3,2879E-03
# iterácií	1	2	3	4	.....	374
$c^T x - b^T y$	1,8241E+06	1,8294E+06	1,7530E+06	1,6677E+06		9,5463E-03
$c^T x - \hat{\chi}$	3,2513E+05	3,4179E+05	3,3567E+05	3,2407E+05		2,7256E-01
$\hat{\chi} - b^T y$	1,4990E+06	1,4876E+06	1,4174E+06	1,3436E+06		7,6370E-03
$\Vert x - \hat{x} \Vert$	2,3762E+03	2,3754E+03	2,3569E+03	2,3220E+03		3,5955E-04
$\Vert y - \hat{y} \Vert$	9,5864E+00	1,0593E+01	1,0996E+01	1,1282E+01		5,9796E-06
$\Vert z - \hat{z} \Vert$	5,9132E+02	6,4127E+02	6,5471E+02	6,6307E+02		3,2676E-04

PRÍKLAD 3	$\varepsilon = $ **1,E-02**	$\lambda = $ **5,E-02**	$\tau = $ **1/2**			
μ	4,8705E+03	4,6269E+03	4,3956E+03	4,1758E+03		9,5486E-05
δ	1,5157E+00	3,8398E-01	2,6413E-02	3,5389E-03		3,2879E-03
# iterácií	1	2	3	4	.....	347
$c^T x - b^T y$	4,2269E+05	4,5311E+05	4,3884E+05	4,1731E+05		9,5423E-03
$c^T x - \hat{\chi}$	7,3472E+04	8,5750E+04	8,5686E+04	8,1903E+04		5,2826E-03
$\hat{\chi} - b^T y$	3,4922E+05	3,6736E+05	3,5315E+05	3,3541E+05		7,6338E-03
$\Vert x - \hat{x} \Vert$	6,9970E+02	7,3875E+02	7,3921E+02	7,3544E+02		6,1722E-04
$\Vert y - \hat{y} \Vert$	9,0375E+00	9,6709E+00	9,5773E+00	9,4883E+00		5,6070E-06
$\Vert z - \hat{z} \Vert$	5,5859E+02	5,7698E+02	5,6415E+02	5,5536E+02		4,1616E-04

PRÍKLAD 4	$\varepsilon = 1,E\text{-}02$	$\lambda = 5,E\text{-}02$	$\tau = 1/2$			
μ	2,1543E+03	2,0466E+03	1,9443E+03	1,8471E+03		9,5963E-05
δ	1,1629E+00	2,9319E-01	1,8914E-02	3,9980E-03		3,2879E-03
# iterácií	1	2	3	4	.....	331
$c^T x - b^T y$	1,9666E+05	2,0216E+05	1,9419E+05	1,8459E+05		9,5900E-03
$c^T x - \hat{\chi}$	3,3256E+04	3,4224E+04	3,3714E+04	3,2600E+04		1,9135E-03
$\hat{\chi} - b^T y$	1,6341E+05	1,6794E+05	1,6048E+05	1,5199E+05		7,6720E-03
$\Vert x - \hat{x} \Vert$	3,7941E+02	3,8365E+02	3,8522E+02	3,8079E+02		1,7919E-04
$\Vert y - \hat{y} \Vert$	5,3119E+00	5,3584E+00	5,3972E+00	5,3206E+00		2,2455E-06
$\Vert z - \hat{z} \Vert$	4,3066E+02	4,4251E+02	4,3845E+02	4,2788E+02		1,1319E-04

PRÍKLAD 5	$\varepsilon = 1,E\text{-}02$	$\lambda = 5,E\text{-}02$	$\tau = 1/2$			
μ	5,0854E+03	4,8311E+03	4,5895E+03	4,3601E+03		9,9699E-05
δ	1,1949E+00	3,4017E-01	2,1896E-02	3,9166E-03		3,2879E-03
# iterácií	1	2	3	4	.....	347
$c^T x - b^T y$	4,6850E+05	4,7672E+05	4,5826E+05	4,3573E+05		9,9633E-03
$c^T x - \hat{\chi}$	6,5542E+04	6,9342E+04	6,8198E+04	6,6273E+04		2,1525E-03
$\hat{\chi} - b^T y$	4,0296E+05	4,0738E+05	3,9006E+05	3,6946E+05		7,9706E-03
$\Vert x - \hat{x} \Vert$	7,0768E+02	7,0158E+02	6,8966E+02	6,7947E+02		1,5653E-04
$\Vert y - \hat{y} \Vert$	3,4889E+00	3,1989E+00	2,9290E+00	2,7912E+00		1,6267E-06
$\Vert z - \hat{z} \Vert$	3,4416E+02	3,4882E+02	3,3071E+02	3,1357E+02		6,9116E-05

Tabuľky 4.3 – 4.7 Namerané hodnoty odchýlok riešení použitím metódy s novou voľbou smeru

Metóda s novou triedou hľadania smerov nám dala úplne iné možnosti analýzy výsledkov. Upustili sme od sledovania zmien odchýlok po zmene parametrov. Vhodnejšie sa javilo overovať, či daný algoritmus skutočne spĺňa naznačené prepojenia s metódami typu „prediktor-korektor“ (uvedené v kapitole 3.3.1).

Výsledky riešenia sady piatich generovaných príkladov nám naznačili, že prepojenie existuje. Dospeli sme k zaujímavým zisteniam. Keďže štartovací bod (vytvorený generátorom spolu s úlohou) je ďalej od centrálnej trajektórie ako nami zvolená presnosť $\tau = 0{,}5$, tak v prvej iterácii prevláda sila korektor zložky, ktorá má za cieľ udržiavanie riešení blízko centrálnej trajektórie. Experimenty ukázali, že už po prvom kroku sa riešenie dostane do jej τ-blízkosti a počas celého procesu ju sleduje (po troch iteráciach s presnosťou rádovo do 10^{-3}, ktorá približne korešponduje s nami preferovanou toleranciou aproximácie centrálnej trajektórie pre prvé dve testované metódy). Avšak toto má za následok vzdialenie bodu od optimálneho riešenia (nárast veľkosti duálnej medzery). No v ďalších krokoch už prediktor komponent zabezpečuje zmenšovanie duálnej medzery a teda lepšiu aproximáciu optima.

4.4. Porovnanie metód

V prvej fáze numerických experimentov sme sa zapodievali vhodnou voľbou parametrov pre klasické metódy (logaritmickú, nelogaritmickú). Teraz ich použijeme pri porovnaní účinnosti metód s metódou s novou triedou hľadania smerov pri riešení súboru 20 vygenerovaných úloh (stále rovnakých rozmerov 80×100). Kompletné tabuľky sa opäť nachádzajú v prílohe.

PRIEM.

	VON.IT.	NEW.IT.	μ_0	θ	$c^T x - b^T y$	$c^T x - \hat{\chi}$	$\hat{\chi} - b^T y$	$\| x - \hat{x} \|$	$\| y - \hat{y} \|$	$\| z - \hat{z} \|$
log	4	15,4	7931,59	0,0050	0,100088	0,020777	0,079311	0,004169	0,000042	0,002437
nelog	3	27,15	13685,61	0,0001	0,056275	0,050037	0,006239	0,001424	0,000005	0,000265
newclass	-	352,25	7535,01	0,9500	0,009746	0,044111	0,007797	0,000387	0,000004	0,00021

Tabuľky 4.8 Priemerné namerané hodnoty odchýlok riešení použitím všetkých troch metód

Z pohľadu najlepších aproximácií v priemerných hodnotách (viď tabuľka) jednoznačne dominuje newclass - metóda s novou triedou hľadania smerov, ktorá má najnižšie všetky priemerné hodnoty s výnimkou duálnej odchýlky.

Náhľad zo strany vnímania priemerného počtu Newtonových iterácií potrebných pre jednotlivé metódy na výpočet optimálneho riešenia úlohy nám však predkladá otázku, aká veľká miera kompromisu je nutná medzi voľbou algoritmu s malým počtom iterácií a horšími aproximáciami a veľmi presným algoritmom s omnoho väčším počtom krokov.

Porovnanie zložitosti algoritmov nebolo začlenené medzi ciele tejto diplomovej práce a hlbšia analýza vlastností a odlišností novej metódy už bohužiaľ bolo mimo jej rámec. Tieto témy však otvárajú mnoho ďalších možností k výskumu v teoretickej rovine aj vedeckej praxi.

Záver

Cieľom práce bolo zhrnúť teoretické poznatky o známych primárno-duálnych algoritmoch metód vnútorného bodu na riešenie úloh lineárneho programovania. Spomedzi klasických metód sme si vybrali logaritmickú a všeobecnú nelogaritmickú metódu, ktoré sme v rovine numerického experimentu použili na riešenie súboru vygenerovaných úloh lineárneho programovania. Ako zástupcu zo skupiny novších metód sme si vybrali metódu s novou triedou hľadania smerov, ktorá úzko súvisí s čoraz populárnejšími metódami typu prediktor-korektor.

Experimentálnym porovnaním týchto troch metód sme prišli k záveru, že síce klasické metódy s menším počtom vonkajších aj vnútorných iterácií dosahovali prijateľné výsledky, špeciálne nelogaritmická funkcia pre vhodné voľby parametrov bola ako tak schopným konkurentom tretej navrhnutej metódy. Aproximácie riešení novou metódou (po rádovo desať krát väčšom počte Newtonových krokoch ako pre zvyšné metódy) dosiahla omnoho lepšie výsledky.

Porovnania zložitosti algoritmov alebo rozšírenia analýzy vlastností, predností a aplikácií metód používajúcich nové hľadanie smerov a rovnako aj voľba iných predpisov funkcie ϕ túto metódu necháva čitateľovi otvorený široký okruh otázok, ktoré by nemuselo byť na škodu preskúmať.

Použitá literatúra

[1] DARVAY, Z. (2003): New Interior Point Algorithms in Linear Programming, AMO – Advance Modeling and Optimization, Volume 5, Number 1

[2] GLINEUR, F. (1999): Interior-point methods for linear programming: a guided tour, Technical report, Service de Mathématique et de Recherche Opérationnelle, Faculté Polytechnique de Mons, Belgium

[3] HALICKÁ, M. (2004): Dvadsať rokov moderných metód vnútorného bodu, Pokroky matematiky, fyziky a astronómie, ročník 49, č. 3, 234 – 244

[4] HALICKÁ, M. (2006): Prednášky z lineárneho programovania 2, Fakulta matematiky, fyziky a informatiky Univerzity Komenského, Bratislava

[5] HAMALA, M. (2006): Prednášky z nelineárneho programovania 2, Fakulta matematiky, fyziky a informatiky Univerzity Komenského, Bratislava

[6] HAMALA, M. (1998): Študijné texty k prednáškam z nelineárneho programovania, rukopis v knižnici Fakulty matematiky, fyziky a informatiky Univerzity Komenského, Bratislava

[7] KOLÁTOR, V. (2006): Metódy vnútorného bodu vo finančných modeloch, Diplomová práca

[8] MÁČAJ, T. (1998): Algoritmy primárno-duálnej metódy vnútorného bodu na riešenie úloh lineárneho programovania, Diplomová práca

[9] PLESNÍK, J. – DUPAČOVÁ, J. – VLACH, M. (1990): Lineárne programovanie, Alfa, Bratislava

[10] WRIGHT, S. J. – NOCEDAL, J. (1999): Numerical Optimization, Springer-Verlag, New York

[11] WRIGHT, S. J. (1997). Primal-Dual Interior-Point Methods, Society for Industrial and Applied Mathematics, Philadelphia

Príloha č. 1 – Výsledky numerických experimentov

Logaritmická metóda

PRÍKLAD 1	ε_1 **= 1,E-08**	ε_2 **= 1,E-03**	θ **= 5,E-02**				
μ	1,0129E+04	5,0644E+02	2,5322E+01	1,2661E+00	6,3305E-02	3,1652E-03	
# vonkajších iterácií	1	2	3	4	5	6	
# vnútorných iterácií	3	5	4	4	3	2	21
$c^T x - b^T y$	1,0129E+06	5,0644E+04	2,5322E+03	1,2661E+02	6,3303E+00	3,1648E-01	
$c^T x - \hat{\chi}$	1,2656E+05	1,0483E+04	5,8756E+02	2,5883E+01	1,2674E+00	6,3271E-02	
$\hat{\chi} - b^T y$	8,8631E+05	4,0160E+04	1,9446E+03	1,0073E+02	5,0629E+00	2,5321E-01	
$\lVert x - \hat{x} \rVert$	2,1431E+03	1,2214E+03	5,2389E+02	9,4780E+00	3,5091E-01	1,7206E-02	
$\lVert y - \hat{y} \rVert$	8,0432E+00	1,3364E+00	1,2140E-01	2,9485E-02	1,9878E-03	1,0064E-04	
$\lVert z - \hat{z} \rVert$	5,9071E+02	6,2728E+01	7,7347E+00	1,8883E+00	1,2588E-01	6,3703E-03	

	ε_1 **= 1,E-08**	ε_2 **= 1,E-03**	θ **= 5,E-03**		
μ	1,0129E+04	5,0644E+01	2,5322E-01	1,2661E-03	
# vonkajších iterácií	1	2	3	4	
# vnútorných iterácií	3	7	4	2	16
$c^T x - b^T y$	1,0129E+06	5,0644E+03	2,5322E+01	1,2777E-01	
$c^T x - \hat{\chi}$	1,2656E+05	1,1369E+03	5,0884E+00	2,6480E-02	
$\hat{\chi} - b^T y$	8,8631E+05	3,9275E+03	2,0234E+01	1,0129E-01	
$\lVert x - \hat{x} \rVert$	2,1431E+03	8,3652E+02	1,4833E+00	6,9300E-03	
$\lVert y - \hat{y} \rVert$	8,0432E+00	1,9153E-01	7,5932E-03	4,0347E-05	
$\lVert z - \hat{z} \rVert$	5,9071E+02	1,1206E+01	4,8165E-01	2,5537E-03	

	ε_1 **= 1,E-08**	ε_2 **= 1,E-03**	θ **= 5,E-04**	
μ	1,0129E+04	5,0644E+00	2,5322E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	9	3	15
$c^T x - b^T y$	1,0129E+06	5,0644E+02	2,5237E-01	
$c^T x - \hat{\chi}$	1,2656E+05	1,0820E+02	4,9792E-02	
$\hat{\chi} - b^T y$	8,8631E+05	3,9823E+02	2,0257E-01	
$\lVert x - \hat{x} \rVert$	2,1431E+03	6,0157E+01	1,3708E-02	
$\lVert y - \hat{y} \rVert$	8,0432E+00	6,1941E-02	8,0563E-05	
$\lVert z - \hat{z} \rVert$	5,9071E+02	4,0606E+00	5,0994E-03	

PRÍKLAD 2

	$\varepsilon_1 = $ **1,E-08**	$\varepsilon_2 = $ **1,E-03**	$\theta = $ **5,E-02**				
μ	2,0488E+04	1,0244E+03	5,1219E+01	2,5610E+00	1,2805E-01	6,4024E-03	
# vonkajších iterácií	1	2	3	4	5	6	
# vnútorných iterácií	2	5	4	3	4	2	20
$c^T x - b^T y$	2,0489E+06	1,0244E+05	5,1219E+03	2,5609E+02	1,2805E+01	6,4040E-01	
$c^T x - \hat{\chi}$	3,7781E+05	2,2769E+04	9,9203E+02	5,0388E+01	2,5586E+00	1,2820E-01	
$\hat{\chi} - b^T y$	1,6710E+06	7,9669E+04	4,1299E+03	2,0570E+02	1,0246E+01	5,1220E-01	
$\lVert x - \hat{x} \rVert$	2,4715E+03	1,9279E+02	1,2710E+01	6,5353E+00	4,7045E-01	2,4169E-02	
$\lVert y - \hat{y} \rVert$	1,0245E+01	1,3439E+01	4,5063E+00	1,9283E-01	8,1043E-03	4,0061E-04	
$\lVert z - \hat{z} \rVert$	6,4632E+02	7,4156E+02	2,4537E+02	1,0517E+01	4,4283E-01	2,1892E-02	

	$\varepsilon_1 = $ **1,E-08**	$\varepsilon_2 = $ **1,E-03**	$\theta = $ **5,E-03**		
μ	2,0488E+04	1,0244E+02	5,1219E-01	2,5610E-03	
# vonkajších iterácií	1	2	3	4	
# vnútorných iterácií	2	6	4	2	14
$c^T x - b^T y$	2,0489E+06	1,0244E+04	5,1237E+01	2,7314E-01	
$c^T x - \hat{\chi}$	3,7781E+05	2,0326E+03	1,0222E+01	6,8310E-02	
$\hat{\chi} - b^T y$	1,6710E+06	8,2113E+03	4,1015E+01	2,0483E-01	
$\lVert x - \hat{x} \rVert$	2,4715E+03	1,1281E+01	1,7743E+00	9,5363E-03	
$\lVert y - \hat{y} \rVert$	1,0245E+01	7,0871E+00	3,3523E-02	1,6077E-04	
$\lVert z - \hat{z} \rVert$	6,4632E+02	3,8670E+02	1,8310E+00	8,7846E-03	

	$\varepsilon_1 = $ **1,E-08**	$\varepsilon_2 = $ **1,E-03**	$\theta = $ **5,E-04**	
μ	2,0488E+04	1,0244E+01	5,1219E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	2	7	5	14
$c^T x - b^T y$	2,0489E+06	1,0244E+03	5,1581E-01	
$c^T x - \hat{\chi}$	3,7781E+05	1,9769E+02	1,0614E-01	
$\hat{\chi} - b^T y$	1,6710E+06	8,2668E+02	4,0967E-01	
$\lVert x - \hat{x} \rVert$	2,4715E+03	1,1750E+01	1,9196E-02	
$\lVert y - \hat{y} \rVert$	1,0245E+01	9,2614E-01	3,2068E-04	
$\lVert z - \hat{z} \rVert$	6,4632E+02	5,0425E+01	1,7525E-02	

PRÍKLAD 3

	$\varepsilon_1 = 1,E\text{-}08$	$\varepsilon_2 = 1,E\text{-}03$	$\theta = 5,E\text{-}02$				
μ	5,1268E+03	2,5634E+02	1,2817E+01	6,4085E-01	3,2043E-02	1,6021E-03	
# vonkajších iterácií	1	2	3	4	5	6	
# vnútorných iterácií	3	4	3	3	3	2	18
$c^T x - b^T y$	5,1268E+05	2,5634E+04	1,2814E+03	6,4087E+01	3,2043E+00	1,6025E-01	
$c^T x - \hat{\chi}$	9,8944E+04	5,0184E+03	2,8810E+02	1,2979E+01	6,4126E-01	3,2084E-02	
$\hat{\chi} - b^T y$	4,1374E+05	2,0616E+04	9,9328E+02	5,1108E+01	2,5631E+00	1,2817E-01	
$\Vert x - \hat{x} \Vert$	7,5408E+02	6,1688E+02	2,8526E+02	5,1637E+00	2,0960E-01	1,0391E-02	
$\Vert y - \hat{y} \Vert$	9,7760E+00	1,4433E+00	1,2781E-01	3,3941E-02	1,8746E-03	9,3986E-05	
$\Vert z - \hat{z} \Vert$	5,8860E+02	8,2438E+01	9,9980E+00	2,5232E+00	1,3915E-01	6,9760E-03	

	$\varepsilon_1 = 1,E\text{-}08$	$\varepsilon_2 = 1,E\text{-}03$	$\theta = 5,E\text{-}03$		
μ	5,1268E+03	2,5634E+01	1,2817E-01	6,4085E-04	
# vonkajších iterácií	1	2	3	4	
# vnútorných iterácií	3	6	4	2	15
$c^T x - b^T y$	5,1268E+05	2,5634E+03	1,2817E+01	6,4236E-02	
$c^T x - \hat{\chi}$	9,8944E+04	5,5459E+02	2,5693E+00	1,2967E-02	
$\hat{\chi} - b^T y$	4,1374E+05	2,0088E+03	1,0248E+01	5,1269E-02	
$\Vert x - \hat{x} \Vert$	7,5408E+02	4,3005E+02	8,6593E-01	4,1486E-03	
$\Vert y - \hat{y} \Vert$	9,7760E+00	1,4902E-01	7,3912E-03	3,7696E-05	
$\Vert z - \hat{z} \Vert$	5,8860E+02	1,1207E+01	5,4874E-01	2,7978E-03	

	$\varepsilon_1 = 1,E\text{-}08$	$\varepsilon_2 = 1,E\text{-}03$	$\theta = 5,E\text{-}04$	
μ	5,1268E+03	2,5634E+00	1,2817E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	7	4	14
$c^T x - b^T y$	5,1268E+05	2,5634E+02	1,2725E-01	
$c^T x - \hat{\chi}$	9,8944E+04	5,4191E+01	2,4714E-02	
$\hat{\chi} - b^T y$	4,1374E+05	2,0215E+02	1,0253E-01	
$\Vert x - \hat{x} \Vert$	7,5408E+02	3,5189E+01	8,2943E-03	
$\Vert y - \hat{y} \Vert$	9,7760E+00	9,2100E-02	7,5313E-05	
$\Vert z - \hat{z} \Vert$	5,8860E+02	6,8966E+00	5,5899E-03	

PRÍKLAD 4

$\varepsilon_1 = 1{,}E\text{-}08$	$\varepsilon_2 = 1{,}E\text{-}03$	$\theta = 5{,}E\text{-}02$					
μ	2,2677E+03	1,1339E+02	5,6693E+00	2,8347E-01	1,4173E-02	7,0867E-04	
# vonkajších iterácií	1	2	3	4	5	6	
# vnútorných iterácií	3	5	4	3	2	2	19
$c^T x - b^T y$	2,2677E+05	1,1339E+04	5,6693E+02	2,8347E+01	1,4173E+00	7,0865E-02	
$c^T x - \hat{\chi}$	3,7269E+04	2,4054E+03	1,1417E+02	5,6715E+00	2,8347E-01	1,4172E-02	
$\hat{\chi} - b^T y$	1,8950E+05	8,9333E+03	4,5276E+02	2,2675E+01	1,1339E+00	5,6693E-02	
$\Vert x - \hat{x} \Vert$	4,0145E+02	1,0619E+02	9,9648E+00	5,2824E-01	2,6480E-02	1,3260E-03	
$\Vert y - \hat{y} \Vert$	5,7280E+00	1,7443E+00	1,2707E-01	6,6234E-03	3,3171E-04	1,6594E-05	
$\Vert z - \hat{z} \Vert$	4,7875E+02	1,0568E+02	6,4388E+00	3,3393E-01	1,6721E-02	8,3642E-04	

$\varepsilon_1 = 1{,}E\text{-}08$	$\varepsilon_2 = 1{,}E\text{-}03$	$\theta = 5{,}E\text{-}03$			
μ	2,2677E+03	1,1339E+01	5,6693E-02	2,8347E-04	
# vonkajších iterácií	1	2	3	4	
# vnútorných iterácií	3	7	3	4	17
$c^T x - b^T y$	2,2677E+05	1,1338E+03	5,6697E+00	2,8352E-02	
$c^T x - \hat{\chi}$	3,7269E+04	2,2984E+02	1,1342E+00	5,6748E-03	
$\hat{\chi} - b^T y$	1,8950E+05	9,0401E+02	4,5354E+00	2,2677E-02	
$\Vert x - \hat{x} \Vert$	4,0145E+02	1,8716E+01	1,0592E-01	5,2880E-04	
$\Vert y - \hat{y} \Vert$	5,7280E+00	2,4234E-01	1,3268E-03	6,6289E-06	
$\Vert z - \hat{z} \Vert$	4,7875E+02	1,2354E+01	6,6881E-02	3,3419E-04	

$\varepsilon_1 = 1{,}E\text{-}08$	$\varepsilon_2 = 1{,}E\text{-}03$	$\theta = 5{,}E\text{-}04$		
μ	2,2677E+03	1,1339E+00	5,6693E-04	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	3	14
$c^T x - b^T y$	2,2677E+05	1,1339E+02	5,6701E-02	
$c^T x - \hat{\chi}$	3,7269E+04	2,2711E+01	1,1343E-02	
$\hat{\chi} - b^T y$	1,8950E+05	9,0675E+01	4,5358E-02	
$\Vert x - \hat{x} \Vert$	4,0145E+02	2,0965E+00	1,0619E-03	
$\Vert y - \hat{y} \Vert$	5,7280E+00	2,6324E-02	1,3276E-05	
$\Vert z - \hat{z} \Vert$	4,7875E+02	1,3282E+00	6,6933E-04	

PRÍKLAD 5

	$\varepsilon_1 = 1$,E-08	$\varepsilon_2 = 1$,E-03	$\theta = 5$,E-02				
μ	5,3530E+03	2,6765E+02	1,3383E+01	6,6913E-01	3,3456E-02	1,6728E-03	
# vonkajších iterácií	1	2	3	4	5	6	
# vnútorných iterácií	3	5	4	3	2	2	19
$c^T x - b^T y$	5,3530E+05	2,6765E+04	1,3383E+03	6,6913E+01	3,3457E+00	1,6731E-01	
$c^T x - \hat{\chi}$	7,4432E+04	5,6660E+03	2,6881E+02	1,3387E+01	6,6917E-01	3,3484E-02	
$\hat{\chi} - b^T y$	4,6087E+05	2,1099E+04	1,0694E+03	5,3526E+01	2,6765E+00	1,3383E-01	
$\lVert x - \hat{x} \rVert$	7,1890E+02	1,4854E+02	1,7325E+01	1,0422E+00	5,2552E-02	2,6261E-03	
$\lVert y - \hat{y} \rVert$	3,4638E+00	1,5119E+00	1,8744E-01	1,0821E-02	5,4563E-04	2,7330E-05	
$\lVert z - \hat{z} \rVert$	3,9124E+02	6,3373E+01	7,8152E+00	4,5920E-01	2,3180E-02	1,1613E-03	

	$\varepsilon_1 = 1$,E-08	$\varepsilon_2 = 1$,E-03	$\theta = 5$,E-03		
μ	5,3530E+03	2,6765E+01	1,3383E-01	6,6913E-04	
# vonkajších iterácií	1	2	3	4	
# vnútorných iterácií	3	7	3	3	16
$c^T x - b^T y$	5,3530E+05	2,6765E+03	1,3383E+01	6,6965E-02	
$c^T x - \hat{\chi}$	7,4432E+04	5,3930E+02	2,6771E+00	1,3433E-02	
$\hat{\chi} - b^T y$	4,6087E+05	2,1372E+03	1,0706E+01	5,3532E-02	
$\lVert x - \hat{x} \rVert$	7,1890E+02	2,8463E+01	2,1043E-01	1,0528E-03	
$\lVert y - \hat{y} \rVert$	3,4638E+00	3,3793E-01	2,1805E-03	1,0972E-05	
$\lVert z - \hat{z} \rVert$	3,9124E+02	1,3945E+01	9,2622E-02	4,6641E-04	

	$\varepsilon_1 = 1$,E-08	$\varepsilon_2 = 1$,E-03	$\theta = 5$,E-04	
μ	5,3530E+03	2,6765E+00	1,3383E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	3	14
$c^T x - b^T y$	5,3530E+05	2,6765E+02	1,3369E-01	
$c^T x - \hat{\chi}$	7,4432E+04	5,3593E+01	2,6620E-02	
$\hat{\chi} - b^T y$	4,6087E+05	2,1406E+02	1,0706E-01	
$\lVert x - \hat{x} \rVert$	7,1890E+02	4,0573E+00	2,1095E-03	
$\lVert y - \hat{y} \rVert$	3,4638E+00	4,2125E-02	2,1843E-05	
$\lVert z - \hat{z} \rVert$	3,9124E+02	1,7815E+00	9,2803E-04	

Nelogaritmická metóda

PRÍKLAD 1	$\varepsilon_1 =$ **1,E-08**	$\varepsilon_2 =$ **1,E-03**			$\varepsilon_1 =$ **1,E-08**	$\varepsilon_2 =$ **1,E-03**		
	$\theta =$ **1,E-03**	$\kappa =$ **1/3**	$\omega =$ **- 1/2**		$\theta =$ **1,E-03**	$\kappa =$ **1/10**	$\omega =$ **- 1/9**	
μ	1,5468E+03	1,5468E+00	1,5468E-03		5,7063E+03	5,7063E+00	5,7063E-03	
# vonkajších iterácií	1	2	3		1	2	3	
# vnútorných iterácií	3	9	29	41	3	8	4	15
$c^T x - b^T y$	1,0276E+06	9,1532E+02	9,0958E-01		1,0017E+06	9,1199E+02	8,6589E-01	
$c^T x - \hat{\chi}$	7,0134E+04	8,6949E+00	7,7930E-04		1,0980E+05	9,9305E+01	4,0057E-02	
$\hat{\chi} - b^T y$	9,5745E+05	9,0662E+02	9,0880E-01		8,9187E+05	8,1269E+02	8,2583E-01	
$\lVert x - \hat{x} \rVert$	1,8195E+03	2,7023E+01	1,6974E-04		2,0457E+03	1,1168E+02	1,4965E-02	
$\lVert y - \hat{y} \rVert$	4,1316E+00	7,0185E-02	9,8290E-05		6,5013E+00	7,9260E-02	2,2104E-04	
$\lVert z - \hat{z} \rVert$	3,6283E+02	4,5837E+00	6,3352E-03		4,9703E+02	5,2194E+00	1,4076E-02	

	$\varepsilon_1 =$ **1,E-08**	$\varepsilon_2 =$ **1,E-03**			$\varepsilon_1 =$ **1,E-08**	$\varepsilon_2 =$ **1,E-03**		
	$\theta =$ **1,E-04**	$\kappa =$ **- 1/3**	$\omega =$ **1/4**		$\theta =$ **1,E-04**	$\kappa =$ **- 1/10**	$\omega =$ **1/11**	
μ	7,1419E+04	7,1419E+00	7,1419E-04		1,8096E+04	1,8096E+00	1,8096E-04	
# vonkajších iterácií	1	2	3		1	2	3	
# vnútorných iterácií	4	10	5	19	3	9	17	29
$c^T x - b^T y$	1,1349E+06	3,6667E+02	1,4325E-01		1,0358E+06	1,3287E+02	1,1187E-01	
$c^T x - \hat{\chi}$	1,7652E+05	2,8768E+02	1,3486E-01		1,4272E+05	5,3281E+01	1,0381E-01	
$\hat{\chi} - b^T y$	9,5838E+05	7,8992E+01	8,3908E-03		8,9307E+05	7,9593E+01	8,0536E-03	
$\lVert x - \hat{x} \rVert$	2,4331E+03	4,4982E+01	3,4941E-02		2,2374E+03	1,4145E+01	2,2759E-03	
$\lVert y - \hat{y} \rVert$	1,6181E+01	3,3358E-02	1,2692E-05		9,9383E+00	2,9257E-02	4,7943E-06	
$\lVert z - \hat{z} \rVert$	1,0942E+03	2,1620E+00	7,8889E-04		7,0764E+02	1,8731E+00	3,0164E-04	

PRÍKLAD 2

	$\varepsilon_1 =$ 1,E-08	**$\varepsilon_2 =$ 1,E-03**		
	$\theta =$ 1,E-03	**$\kappa =$ 1/3**	**$\omega =$ - 1/2**	
μ	2,7079E+03	2,7079E+00	2,7079E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	8	19
$c^T x - b^T y$	2,0992E+06	1,9095E+03	1,9042E+00	
$c^T x - \hat{\chi}$	1,6978E+05	5,1160E+00	1,7750E-04	
$\hat{\chi} - b^T y$	1,9294E+06	1,9044E+03	1,9040E+00	
$\| x - \hat{x} \|$	2,0675E+03	2,0638E+00	1,8608E-04	
$\| y - \hat{y} \|$	1,6734E+01	4,0911E-01	3,9744E-04	
$\| z - \hat{z} \|$	9,4193E+02	2,2417E+01	2,1786E-02	

	$\varepsilon_1 =$ 1,E-08	**$\varepsilon_2 =$ 1,E-03**		
	$\theta =$ 1,E-03	**$\kappa =$ 1/10**	**$\omega =$ - 1/9**	
μ	1,0998E+04	1,0998E+01	1,0998E-02	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	7	6	16
$c^T x - b^T y$	2,0194E+06	1,8328E+03	1,7522E+00	
$c^T x - \hat{\chi}$	3,0927E+05	1,4421E+02	7,0506E-02	
$\hat{\chi} - b^T y$	1,7101E+06	1,6886E+03	1,6817E+00	
$\| x - \hat{x} \|$	2,3326E+03	9,9085E+00	1,9888E-02	
$\| y - \hat{y} \|$	1,2021E+01	1,1379E+00	8,8119E-04	
$\| z - \hat{z} \|$	7,1702E+02	6,2077E+01	4,8209E-02	

	$\varepsilon_1 =$ 1,E-08	**$\varepsilon_2 =$ 1,E-03**		
	$\theta =$ 1,E-04	**$\kappa =$ - 1/3**	**$\omega =$ 1/4**	
μ	1,7207E+05	1,7207E+01	1,7207E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	4	7	6	17
$c^T x - b^T y$	2,3342E+06	8,9463E+02	7,5597E-01	
$c^T x - \hat{\chi}$	6,1811E+05	7,1319E+02	7,3925E-01	
$\hat{\chi} - b^T y$	1,7161E+06	1,8144E+02	1,6722E-02	
$\| x - \hat{x} \|$	2,9731E+03	1,7024E+01	5,0642E-02	
$\| y - \hat{y} \|$	7,8344E+00	1,1343E+00	5,0349E-05	
$\| z - \hat{z} \|$	6,5586E+02	6,1413E+01	2,7401E-03	

	$\varepsilon_1 =$ 1,E-08	**$\varepsilon_2 =$ 1,E-03**		
	$\theta =$ 1,E-04	**$\kappa =$ - 1/10**	**$\omega =$ 1/11**	
μ	3,8512E+04	3,8512E+00	3,8512E-04	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	12	23
$c^T x - b^T y$	2,1065E+06	2,7722E+02	5,1257E-02	
$c^T x - \hat{\chi}$	4,4888E+05	1,1335E+02	3,5050E-02	
$\hat{\chi} - b^T y$	1,6577E+06	1,6387E+02	1,6207E-02	
$\| x - \hat{x} \|$	2,6216E+03	9,6977E+00	3,4396E-03	
$\| y - \hat{y} \|$	8,6680E+00	2,6360E-01	1,9052E-05	
$\| z - \hat{z} \|$	5,9795E+02	1,4343E+01	1,0399E-03	

PRÍKLAD 3

	$\varepsilon_1 = 1{,}E\text{-}08$	$\varepsilon_2 = 1{,}E\text{-}03$		
	$\theta = 1{,}E\text{-}03$	$\kappa = 1/3$	$\omega = -\ 1/2$	
μ	1,1934E+03	1,1934E+00	1,1934E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	4	8	21	33
$c^T x - b^T y$	6,0833E+05	5,5910E+02	5,5613E-01	
$c^T x - \hat{\chi}$	4,1171E+04	5,4657E+00	3,8669E-04	
$\hat{\chi} - b^T y$	5,6716E+05	5,5364E+02	5,5574E-01	
$\lVert x - \hat{x} \rVert$	7,0157E+02	2,7285E+01	1,0316E-04	
$\lVert y - \hat{y} \rVert$	1,1574E+01	1,3135E-01	1,6850E-04	
$\lVert z - \hat{z} \rVert$	6,9040E+02	9,8704E+00	1,2590E-02	

	$\varepsilon_1 = 1{,}E\text{-}08$	$\varepsilon_2 = 1{,}E\text{-}03$		
	$\theta = 1{,}E\text{-}03$	$\kappa = 1/10$	$\omega = -\ 1/9$	
μ	3,2601E+03	3,2601E+00	3,2601E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	7	4	14
$c^T x - b^T y$	5,2678E+05	4,8136E+02	4,5883E-01	
$c^T x - \hat{\chi}$	7,9680E+04	4,7815E+01	1,8008E-02	
$\hat{\chi} - b^T y$	4,4710E+05	4,3354E+02	4,4082E-01	
$\lVert x - \hat{x} \rVert$	7,3732E+02	7,5783E+01	9,0673E-03	
$\lVert y - \hat{y} \rVert$	1,0273E+01	1,2271E-01	2,4908E-04	
$\lVert z - \hat{z} \rVert$	6,1447E+02	9,2692E+00	1,8522E-02	

	$\varepsilon_1 = 1{,}E\text{-}08$	$\varepsilon_2 = 1{,}E\text{-}03$		
	$\theta = 1{,}E\text{-}04$	$\kappa = -\ 1/3$	$\omega = 1/4$	
μ	2,5041E+04	2,5041E+00	2,5041E-04	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	7	18
$c^T x - b^T y$	5,2199E+05	2,1236E+02	3,3968E-01	
$c^T x - \hat{\chi}$	1,7182E+05	1,7753E+02	3,3608E-01	
$\hat{\chi} - b^T y$	3,5017E+05	3,4829E+01	3,5966E-03	
$\lVert x - \hat{x} \rVert$	8,1676E+02	2,4227E+01	2,0160E-02	
$\lVert y - \hat{y} \rVert$	8,5527E+00	3,5797E-02	6,3769E-06	
$\lVert z - \hat{z} \rVert$	5,3178E+02	2,6613E+00	4,7052E-04	

	$\varepsilon_1 = 1{,}E\text{-}08$	$\varepsilon_2 = 1{,}E\text{-}03$		
	$\theta = 1{,}E\text{-}04$	$\kappa = -\ 1/10$	$\omega = 1/11$	
μ	8,1565E+03	8,1565E-01	8,1565E-05	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	12	23
$c^T x - b^T y$	5,0759E+05	6,7155E+01	8,4151E-03	
$c^T x - \hat{\chi}$	1,1956E+05	2,8636E+01	4,5384E-03	
$\hat{\chi} - b^T y$	3,8802E+05	3,8519E+01	3,8767E-03	
$\lVert x - \hat{x} \rVert$	7,7183E+02	7,6203E+00	1,4779E-03	
$\lVert y - \hat{y} \rVert$	9,3335E+00	3,1279E-02	3,7091E-06	
$\lVert z - \hat{z} \rVert$	5,6672E+02	2,3230E+00	2,7478E-04	

PRÍKLAD 4

	ε_1 = 1,E-08	ε_2 = 1,E-03		
	θ = 1,E-03	κ = 1/3	ω = - 1/2	
μ	6,0044E+02	6,0044E-01	6,0044E-04	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	4	9	31	44
$c^T x - b^T y$	2,3799E+05	2,1021E+02	2,0926E-01	
$c^T x - \hat{\chi}$	2,2285E+04	1,0050E+00	5,0575E-05	
$\hat{\chi} - b^T y$	2,1570E+05	2,0921E+02	2,0920E-01	
$\lVert x - \hat{x} \rVert$	3,3267E+02	2,9052E-01	9,4508E-06	
$\lVert y - \hat{y} \rVert$	5,2558E+00	2,9438E-02	2,9441E-05	
$\lVert z - \hat{z} \rVert$	4,6086E+02	1,4910E+00	1,4909E-03	

	ε_1 = 1,E-08	ε_2 = 1,E-03		
	θ = 1,E-03	κ = 1/10	ω = - 1/9	
μ	1,5115E+03	1,5115E+00	1,5115E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	5	16
$c^T x - b^T y$	2,2779E+05	2,0602E+02	1,9587E-01	
$c^T x - \hat{\chi}$	3,2965E+04	1,8939E+01	8,7265E-03	
$\hat{\chi} - b^T y$	1,9482E+05	1,8708E+02	1,8715E-01	
$\lVert x - \hat{x} \rVert$	3,8172E+02	2,2575E+00	1,0713E-03	
$\lVert y - \hat{y} \rVert$	5,4466E+00	4,4301E-02	4,4580E-05	
$\lVert z - \hat{z} \rVert$	4,6549E+02	2,2335E+00	2,2448E-03	

	ε_1 = 1,E-08	ε_2 = 1,E-03		
	θ = 1,E-04	κ = - 1/3	ω = 1/4	
μ	9,0686E+03	9,0686E-01	9,0686E-05	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	10	4	17
$c^T x - b^T y$	2,3459E+05	7,8673E+01	6,0419E-02	
$c^T x - \hat{\chi}$	5,0288E+04	6,1164E+01	5,8664E-02	
$\hat{\chi} - b^T y$	1,8430E+05	1,7509E+01	1,7551E-03	
$\lVert x - \hat{x} \rVert$	4,6189E+02	3,0977E+00	3,0963E-03	
$\lVert y - \hat{y} \rVert$	7,6653E+00	9,3929E-03	9,6245E-07	
$\lVert z - \hat{z} \rVert$	5,7623E+02	4,7788E-01	4,8944E-05	

	ε_1 = 1,E-08	ε_2 = 1,E-03		
	θ = 1,E-04	κ = - 1/10	ω = 1/11	
μ	3,4198E+03	3,4198E-01	3,4198E-05	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	9	5	17
$c^T x - b^T y$	2,2741E+05	2,9336E+01	3,6695E-03	
$c^T x - \hat{\chi}$	4,1392E+04	1,1580E+01	1,8933E-03	
$\hat{\chi} - b^T y$	1,8602E+05	1,7756E+01	1,7762E-03	
$\lVert x - \hat{x} \rVert$	4,2046E+02	8,6454E-01	1,9984E-04	
$\lVert y - \hat{y} \rVert$	6,1435E+00	6,3019E-03	6,3360E-07	
$\lVert z - \hat{z} \rVert$	4,9913E+02	3,1841E-01	3,2003E-05	

PRÍKLAD 5

	$\varepsilon_1 = 1,E\text{-}08$	$\varepsilon_2 = 1,E\text{-}03$		
	$\theta = 1,E\text{-}03$	$\kappa = 1/3$	$\omega = -\ 1/2$	
μ	1,0477E+03	1,0477E+00	1,0477E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	4	9	13	26
$c^T x - b^T y$	5,3424E+05	4,7399E+02	4,7160E-01	
$c^T x - \hat{\chi}$	4,6315E+04	2,4623E+00	5,3674E-05	
$\hat{\chi} - b^T y$	4,8793E+05	4,7153E+02	4,7154E-01	
$\Vert x - \hat{x} \Vert$	5,8661E+02	7,1518E-01	2,4938E-05	
$\Vert y - \hat{y} \Vert$	2,8517E+00	4,0484E-02	4,0513E-05	
$\Vert z - \hat{z} \Vert$	3,6159E+02	1,6706E+00	1,6718E-03	

	$\varepsilon_1 = 1,E\text{-}08$	$\varepsilon_2 = 1,E\text{-}03$		
	$\theta = 1,E\text{-}03$	$\kappa = 1/10$	$\omega = -\ 1/9$	
μ	3,2606E+03	3,2606E+00	3,2606E-03	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	5	16
$c^T x - b^T y$	5,3036E+05	4,8192E+02	4,5780E-01	
$c^T x - \hat{\chi}$	6,6594E+04	4,5163E+01	2,0915E-02	
$\hat{\chi} - b^T y$	4,6377E+05	4,3676E+02	4,3689E-01	
$\Vert x - \hat{x} \Vert$	6,8153E+02	4,5785E+00	2,3147E-03	
$\Vert y - \hat{y} \Vert$	2,9908E+00	6,6682E-02	6,8581E-05	
$\Vert z - \hat{z} \Vert$	3,7325E+02	2,7841E+00	2,8736E-03	

	$\varepsilon_1 = 1,E\text{-}08$	$\varepsilon_2 = 1,E\text{-}03$		
	$\theta = 1,E\text{-}04$	$\kappa = -\ 1/3$	$\omega = 1/4$	
μ	2,8715E+04	2,8715E+00	2,8715E-04	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	4	8	6	18
$c^T x - b^T y$	5,7523E+05	1,8156E+02	1,8935E-01	
$c^T x - \hat{\chi}$	9,6513E+04	1,3952E+02	1,8513E-01	
$\hat{\chi} - b^T y$	4,7872E+05	4,2041E+01	4,2268E-03	
$\Vert x - \hat{x} \Vert$	8,2691E+02	4,9801E+00	4,9374E-03	
$\Vert y - \hat{y} \Vert$	6,6958E+00	1,9135E-02	2,1810E-06	
$\Vert z - \hat{z} \Vert$	5,1804E+02	8,4830E-01	9,7751E-05	

	$\varepsilon_1 = 1,E\text{-}08$	$\varepsilon_2 = 1,E\text{-}03$		
	$\theta = 1,E\text{-}04$	$\kappa = -\ 1/10$	$\omega = 1/11$	
μ	8,8260E+03	8,8260E-01	8,8260E-05	
# vonkajších iterácií	1	2	3	
# vnútorných iterácií	3	8	5	16
$c^T x - b^T y$	5,4362E+05	6,9306E+01	8,7452E-03	
$c^T x - \hat{\chi}$	8,1714E+04	2,7028E+01	4,5158E-03	
$\hat{\chi} - b^T y$	4,6190E+05	4,2278E+01	4,2294E-03	
$\Vert x - \hat{x} \Vert$	7,5407E+02	1,5925E+00	3,7019E-04	
$\Vert y - \hat{y} \Vert$	4,1830E+00	1,1073E-02	1,1316E-06	
$\Vert z - \hat{z} \Vert$	4,1798E+02	4,7689E-01	4,8858E-05	

Druhá fáza numerických experimentov – porovnanie všetkých metód

MET.	VON.IT.	NEW.IT.	μ_0	θ	$c^T x - b^T y$	$c^T x - \hat{\chi}$	$\hat{\chi} - b^T y$	$\lVert x - \hat{x} \rVert$	$\lVert y - \hat{y} \rVert$	$\lVert z - \hat{z} \rVert$
Pr. 1										
log	4	16	1,0129E+04	0,0050	0,127767	0,026480	0,101287	0,006930	0,000040	0,002554
nelog	3	29	1,8096E+04	0,0010	0,111867	0,103814	0,008054	0,002276	0,000005	0,000302
newclass	-	360	9,6223E+03	0,9500	0,009678	0,009378	0,007742	0,000527	0,000003	0,000195
Pr. 2										
log	4	14	2,0488E+04	0,0050	0,273135	0,068310	0,204825	0,009536	0,000161	0,008785
nelog	3	23	3,8512E+04	0,0001	0,051257	0,035050	0,016207	0,003440	0,000019	0,001040
newclass	-	374	1,9463E+04	0,9500	0,009546	0,272557	0,007637	0,000360	0,000006	0,000327
Pr. 3										
log	4	15	5,1268E+03	0,0050	0,064236	0,012967	0,051269	0,004149	0,000038	0,002798
nelog	3	23	8,1565E+03	0,0001	0,008415	0,004538	0,003877	0,001478	0,000004	0,000275
newclass	-	347	4,8705E+03	0,9500	0,009542	0,005283	0,007634	0,000617	0,000006	0,000416
Pr. 4										
log	4	17	2,2677E+03	0,0050	0,028352	0,005675	0,022677	0,000529	0,000007	0,000334
nelog	3	17	3,4198E+03	0,0001	0,003670	0,001893	0,001776	0,000200	0,000001	0,000032
newclass	-	331	2,1543E+03	0,9500	0,009590	0,001914	0,007672	0,000179	0,000002	0,000113
Pr. 5										
log	4	16	5,3530E+03	0,0050	0,066965	0,013433	0,053532	0,001053	0,000011	0,000466
nelog	3	16	8,8260E+03	0,0001	0,008745	0,004516	0,004229	0,000370	0,000001	0,000049
newclass	-	347	5,0854E+03	0,9500	0,009963	0,002152	0,007971	0,000157	0,000002	0,000069
Pr. 6										
log	4	18	6,6799E+03	0,0050	0,083455	0,016638	0,066817	0,001863	0,000031	0,001745
nelog	3	26	1,1210E+04	0,0001	0,014834	0,009547	0,005287	0,000701	0,000003	0,000183
newclass	-	352	6,3459E+03	0,9500	0,009620	0,001739	0,007696	0,000215	0,000004	0,000201
Pr. 7										
log	4	14	6,8945E+03	0,0050	0,086159	0,017224	0,068936	0,007847	0,000021	0,001432
nelog	3	16	1,1485E+04	0,0001	0,016274	0,010815	0,005459	0,002848	0,000002	0,000148
newclass	-	352	6,5497E+03	0,9500	0,009929	0,001221	0,007944	0,000907	0,000002	0,000165
Pr. 8										
log	4	15	5,1034E+03	0,0050	0,064025	0,012989	0,051035	0,000543	0,000011	0,000637
nelog	3	50	8,4135E+03	0,0001	0,010835	0,006824	0,004011	0,000219	0,000001	0,000067
newclass	-	346	4,8483E+03	0,9500	0,009999	0,001992	0,007999	0,000085	0,000002	0,000100
Pr. 9										
log	4	16	7,2815E+03	0,0050	0,091224	0,018412	0,072812	0,002295	0,000036	0,001900
nelog	3	34	1,2348E+04	0,0001	0,008144	0,002317	0,005826	0,000868	0,000004	0,000205
newclass	-	353	6,9174E+03	0,9500	0,009962	0,003356	0,007970	0,000251	0,000004	0,000208

Pr. 10										
log	4	16	5,1256E+03	0,0050	0,064093	0,012829	0,051263	0,001249	0,000018	0,001153
nelog	3	33	8,5413E+03	0,0001	0,013780	0,009699	0,004081	0,000436	0,000002	0,000114
newclass	-	347	4,8693E+03	0,9500	0,009540	0,002387	0,007632	0,000186	0,000003	0,000172
Pr. 11										
log	4	16	1,1134E+04	0,0050	0,139431	0,028094	0,111338	0,007755	0,000122	0,008207
nelog	3	18	1,9720E+04	0,0001	0,612101	0,603184	0,008917	0,001785	0,000013	0,000863
newclass	-	362	1,0577E+04	0,9500	0,009601	0,079810	0,007681	0,000534	0,000008	0,000566
Pr. 12										
log	4	17	4,6759E+03	0,0050	0,058619	0,011862	0,046757	0,002620	0,000013	0,000861
nelog	3	92	7,7245E+03	0,0001	0,012910	0,009239	0,003670	0,001061	0,000001	0,000086
newclass	-	345	4,4421E+03	0,9500	0,009643	0,003525	0,007715	0,000431	0,000002	0,000143
Pr. 13										
log	4	15	6,1467E+03	0,0050	0,077066	0,015561	0,061505	0,002236	0,000025	0,001337
nelog	3	30	1,0142E+04	0,0001	0,010511	0,005732	0,004779	0,000810	0,000003	0,000142
newclass	-	350	5,8394E+03	0,9500	0,009809	0,003024	0,007847	0,000285	0,000003	0,000171
Pr. 14										
log	4	14	2,8376E+03	0,0050	0,035510	0,007133	0,028377	0,001286	0,000004	0,000154
nelog	3	17	4,3607E+03	0,0001	0,003612	0,001402	0,002210	0,000458	0,000000	0,000015
newclass	-	335	2,6957E+03	0,9500	0,009774	0,001940	0,007819	0,000354	0,000001	0,000042
Pr. 15										
log	4	17	9,4626E+03	0,0050	0,118132	0,023505	0,094627	0,010828	0,000015	0,000884
nelog	3	23	1,5888E+04	0,0001	0,007437	0,000148	0,007289	0,003257	0,000002	0,000092
newclass	-	359	8,9895E+03	0,9500	0,009517	0,000542	0,007614	0,000871	0,000001	0,000071
Pr. 16										
log	4	14	5,4403E+03	0,0050	0,068144	0,013709	0,054436	0,003085	0,000054	0,002260
nelog	3	16	8,6824E+03	0,0001	0,121792	0,117521	0,004271	0,001172	0,000005	0,000223
newclass	-	348	5,1682E+03	0,9500	0,009619	0,006080	0,007696	0,000436	0,000008	0,000320
Pr. 17										
log	4	15	1,4012E+04	0,0050	0,175606	0,035485	0,140121	0,004281	0,000028	0,001742
nelog	3	16	2,5046E+04	0,0001	0,043183	0,032198	0,010985	0,001613	0,000003	0,000189
newclass	-	366	1,3311E+04	0,9500	0,009841	0,007286	0,007873	0,000241	0,000002	0,000098
Pr. 18										
log	4	15	1,7400E+04	0,0050	0,216476	0,042581	0,173895	0,008853	0,000129	0,006904
nelog	3	18	3,1987E+04	0,0001	0,034129	0,020381	0,013748	0,003252	0,000015	0,000821
newclass	-	370	1,6530E+04	0,9500	0,009954	0,472690	0,007963	0,000405	0,000006	0,000315
Pr. 19										
log	4	15	8,0471E+03	0,0050	0,100537	0,020091	0,080446	0,005491	0,000059	0,003734

nelog	3	22	1,2966E+04	0,0001	0,022534	0,016418	0,006117	0,001882	0,000006	0,000367
newclass	-	355	7,6448E+03	0,9500	0,009937	0,002387	0,007949	0,000543	0,000006	0,000369
Pr. 20										
log	4	13	5,0268E+03	0,0050	0,062833	0,012561	0,050272	0,000949	0,000018	0,000845
nelog	3	24	8,1876E+03	0,0001	0,009477	0,005499	0,003978	0,000360	0,000002	0,000083
newclass	-	346	4,7754E+03	0,9500	0,009848	0,002958	0,007879	0,000149	0,000003	0,000133
AVG										
metoda	VON.IT.	NEW.IT.	μ_0	θ	$c^T x - b^T y$	$c^T x - \hat{\chi}$	$\hat{\chi} - b^T y$	$\lVert x - \hat{x} \rVert$	$\lVert y - \hat{y} \rVert$	$\lVert z - \hat{z} \rVert$
log	4	15,40	7931,59	0,0050	0,100088	0,020777	0,079311	0,004169	0,000042	0,002437
nelog	3	27,15	13685,61	0,0001	0,056275	0,050037	0,006239	0,001424	0,000005	0,000265
newclass	-	352,25	7535,01	0,9500	0,009746	0,044111	0,007797	0,000387	0,000004	0,000210

Príloha č. 2 - Zdrojový kód programu

Program použitý na generovanie a riešenie úloh lineárneho programovania tromi navrhnutými metódami vnútorného bodu.
Zoznam m-fileov so skriptom:

- main.m
- riesitel_log.m
- riesitel_nelog.m
- riesitel_newclass.m
- generator.m

Zoznam m-fileov s funkciou:

- log_newton_smer(A, x, z, mi)
- log_newton_krok(c, b, x, dx, y, dy, z, dz, mi)
- nelog_newton_smer(A, x, z, mi, h, omega)
- nelog_newton_krok(c, b, x, dx, y, dy, z, dz, mi, h, omega)
- new_class_newton_smer(A, x, z, mi)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% HLAVNY M-FILE
clear all; close all; format long;

generator
load priklad A_mat c_vek b_vek start_y_vek start_z_vek start_x_vek opt_x_vek opt_y_vek
opt_z_vek gama
clc

riesitel_log;
riesitel_nelog;
riesitel_newclass;

load metoda_log log_mi_count log_mi_newton_count log_bl_opt_ries log_dual_medz
log_prim_odch log_dual_odch log_pres_x log_pres_y log_pres_z
load metoda_nelog nelog_mi_count nelog_mi_newton_count nelog_bl_opt_ries
nelog_dual_medz nelog_prim_odch nelog_dual_odch nelog_pres_x nelog_pres_y
nelog_pres_z
load metoda_newclass newclass_mi_count newclass_dual_medz newclass_prim_odch
newclass_dual_odch newclass_pres_x newclass_pres_y newclass_pres_z

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% RIESITEL LOG
```

```
clear all; close all;

load priklad A_mat c_vek b_vek start_x_vek start_y_vek start_z_vek opt_x_vek opt_y_vek
opt_z_vek gama
clc

disp('***********************************')
disp('   LOGARITMICKA BARIEROVA METODA')
disp('***********************************')
disp(' ')

ok=0;
while ok==0;
   ok=1;
   eps1 = input('epsilon 1 (blizkost k optimalnemu rieseniu) z intervalu (0,1) = ');
   if eps1 < 0 || eps1 > 1
      disp('ZADAJ epsilon 1 Z INTERVALU (0,1) ')
      ok=0;
   end
end
ok=0;
while ok==0;
   ok=1;
   eps2 = input('epsilon 2 (blizkost k centralnej trajektorii) z intervalu (0,1) = ');
   if eps2 < 0 || eps2 > 1
      disp('ZADAJ epsilon 2 Z INTERVALU (0,1) ')
      ok=0;
   end
end
ok=0;
while ok==0;
   ok=1;
   theta = input('theta (koeficient redukcie) z intervalu (0,1) = ');
   if theta < 0 || theta > 1
      disp('ZADAJ thetu Z INTERVALU (0,1) ')
      ok=0;
   end
end

% pociatocne vektory
x_vek = start_x_vek;
y_vek = start_y_vek;
z_vek = start_z_vek;
```

```
e_vek = ones(length(x_vek),1);
mi = sum(x_vek .* z_vek) / length(x_vek);
mi_vek = mi * e_vek;
xi_zi_vek = x_vek .* z_vek;

mi_count=0;
% blizkost k optimalnemu rieseniu
bl_opt_ries = norm(c_vek' * x_vek - b_vek' * y_vek) / (1 + norm(c_vek' * x_vek));
while bl_opt_ries > eps1
   % blizkost k centralnej trajektorii
   bl_cen_traj = norm(mi_vek - xi_zi_vek) / norm(mi_vek);
   newton_count=0;
   while bl_cen_traj > eps2
      % smery a krok
      [dx, dy, dz] = log_newton_smer(A_mat, x_vek, z_vek, mi);
      [alfa_opt, beta_opt] = log_newton_krok(c_vek, b_vek, x_vek, dx, y_vek, dy, z_vek,
dz, mi);
      % nova iteracia
      x_vek = x_vek + alfa_opt * dx;
      y_vek = y_vek + beta_opt * dy;
      z_vek = z_vek + beta_opt * dz;
      newton_count = newton_count + 1;
      xi_zi_vek = x_vek .* z_vek;
      bl_cen_traj = norm(mi_vek - xi_zi_vek) / norm(mi_vek);
   end
   mi = theta * mi;
   mi_vek = mi * e_vek;
   bl_opt_ries = norm(c_vek' * x_vek - b_vek' * y_vek) / (1 + norm(c_vek' * x_vek));
   % kolko krat menime mi
   mi_count = mi_count + 1;
   % kolko newton krokov pri fixnom mi
   mi_newton_count(mi_count) = newton_count;
end
log_mi_count = mi_count
log_mi_newton_count = mi_newton_count
log_bl_opt_ries = bl_opt_ries
log_dual_medz = c_vek' * x_vek - b_vek' * y_vek
log_prim_odch = c_vek' * x_vek - gama
log_dual_odch = gama - b_vek' * y_vek
log_pres_x = norm(x_vek - opt_x_vek)
log_pres_y = norm(y_vek - opt_y_vek)
log_pres_z = norm(z_vek - opt_z_vek)
```

```
save metoda_log log_mi_count log_mi_newton_count log_bl_opt_ries log_dual_medz
log_prim_odch log_dual_odch log_pres_x log_pres_y log_pres_z
pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% RIESITEL NELOG
clear all; close all;

load priklad A_mat c_vek b_vek start_x_vek start_y_vek start_z_vek opt_x_vek opt_y_vek
opt_z_vek gama
clc

disp('************************************')
disp('   NELOGARITMICKA BARIEROVA METODA')
disp('************************************')
disp(' ')

ok=0;
while ok==0;
   ok=1;
   eps1 = input('epsilon 1 (blizkost k optimalnemu rieseniu) z intervalu (0,1) = ');
   if eps1 < 0 || eps1 > 1
      disp('ZADAJ epsilon 1 Z INTERVALU (0,1) ')
      ok=0;
   end
end
ok=0;
while ok==0;
   ok=1;
   eps2 = input('epsilon 2 (blizkost k centralnej trajektorii) z intervalu (0,1) = ');
   if eps2 < 0 || eps2 > 1
      disp('ZADAJ epsilon 2 Z INTERVALU (0,1) ')
      ok=0;
   end
end
ok=0;
while ok==0;
   ok=1;
   theta = input('theta (koeficient redukcie) z intervalu (0,1) = ');
   if theta < 0 || theta > 1
      disp('ZADAJ thetu Z INTERVALU (0,1) ')
      ok=0;
   end
```

```
end
ok=0;
while ok==0;
   ok=1;
   h_param = input('hodnota parametra pre nelogaritmicku barierovu fciu h = ');
   if h_param >= 2  || h_param == 0
      disp('ZADAJ parameter h Z INTERVALU (-inf,1)-{0} ')
      ok=0;
   end
end
om_param = 1 - 1 / (1 - h_param);

% pociatocne vektory
x_vek = start_x_vek;
y_vek = start_y_vek;
z_vek = start_z_vek;

e_vek = ones(length(x_vek),1);
mi = sum(start_x_vek.^(1 - h_param) .* start_z_vek) / length(start_x_vek);
mi_vek = mi * e_vek;
xi_zi_vek = start_x_vek .* start_z_vek;

mi_count=0;
% blizkost k optimalnemu rieseniu
bl_opt_ries = norm(c_vek' * x_vek - b_vek' * y_vek) / (1 + norm(c_vek' * x_vek));
while bl_opt_ries > eps1
   % blizkost k centralnej trajektorii
   P_bl_cen_traj = norm(mi_vek - diag(x_vek)^(1 - h_param) * diag(z_vek) * e_vek) /
norm(mi_vek);
   D_bl_cen_traj = norm(mi^(1 - om_param) * e_vek - diag(z_vek)^(1 - om_param) *
diag(x_vek) * e_vek) / norm(mi^(1 - om_param) * e_vek);
   newton_count=0;
   while (P_bl_cen_traj > eps2 || D_bl_cen_traj > eps2)
      % smery a krok
      [dx, dy, dz] = nelog_newton_smer(A_mat, x_vek, z_vek, mi, h_param, om_param);
      [alfa_opt, beta_opt] = nelog_newton_krok(c_vek, b_vek, x_vek, dx, y_vek, dy, z_vek,
dz, mi, h_param, om_param);
      % nova iteracia
      x_vek = x_vek + alfa_opt * dx;
      y_vek = y_vek + beta_opt * dy;
      z_vek = z_vek + beta_opt * dz;
      newton_count = newton_count + 1;
      xi_zi_vek = x_vek .* z_vek;
```

```
      P_bl_cen_traj = norm(mi_vek - diag(x_vek)^(1 - h_param) * diag(z_vek) * e_vek) /
norm(mi_vek);
      D_bl_cen_traj = norm(mi^(1 - om_param) * e_vek - diag(z_vek)^(1 - om_param) *
diag(x_vek) * e_vek) / norm(mi^(1 - om_param) * e_vek);
   end
   mi = theta * mi;
   mi_vek = mi * e_vek;
   bl_opt_ries = norm(c_vek' * x_vek - b_vek' * y_vek) / (1 + norm(c_vek' * x_vek));
   % kolko krat menime mi
   mi_count = mi_count + 1;
   % kolko newton krokov pri fixnom mi
   mi_newton_count(mi_count) = newton_count;
end
nelog_mi_count = mi_count
nelog_mi_newton_count = mi_newton_count
nelog_bl_opt_ries = bl_opt_ries
nelog_dual_medz = c_vek' * x_vek - b_vek' * y_vek
nelog_prim_odch = c_vek' * x_vek - gama
nelog_dual_odch = gama - b_vek' * y_vek
nelog_pres_x = norm(x_vek - opt_x_vek)
nelog_pres_y = norm(y_vek - opt_y_vek)
nelog_pres_z = norm(z_vek - opt_z_vek)

save metoda_nelog nelog_mi_count nelog_mi_newton_count nelog_bl_opt_ries
nelog_dual_medz nelog_prim_odch nelog_dual_odch nelog_pres_x nelog_pres_y
nelog_pres_z
pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% RIESITEL NEWCLASS
clear all; close all;

load priklad A_mat c_vek b_vek start_x_vek start_y_vek start_z_vek opt_x_vek opt_y_vek
opt_z_vek gama
clc

disp('**********************')
disp('   NEW CLASS METODA')
disp('**********************')
disp(' ')

eps=1e-2
lambda=1/(2 * sqrt(length(start_x_vek)))
```

```
tau=1/2

% pociatocne vektory
x_vek = start_x_vek;
y_vek = start_y_vek;
z_vek = start_z_vek;

e_vek = ones(length(x_vek),1);
mi = sum(x_vek .* z_vek) / length(x_vek);
v_vek = sqrt(x_vek .* z_vek / mi);
sigma = norm(e_vek - v_vek);

mi_count=0;
while x_vek' * z_vek > eps
   mi = mi * (1 - lambda);
   % smery
   [dx, dy, dz] = newclass_newton_smer(A_mat, x_vek, z_vek, mi);
   % nova iteracia
   x_vek = x_vek + dx;
   y_vek = y_vek + dy;
   z_vek = z_vek + dz;
   mi_count = mi_count + 1;
   v_vek = sqrt(x_vek .* z_vek / mi);
   sigma = norm(e_vek - v_vek);
   if sigma > tau
      disp('sigma > tau')
      mi_count
      sigma
   end
end
newclass_mi_count = mi_count
newclass_dual_medz = x_vek' * z_vek
newclass_prim_odch = c_vek' * x_vek - gama
newclass_dual_odch = gama - b_vek' * y_vek
newclass_pres_x = norm(x_vek - opt_x_vek)
newclass_pres_y = norm(y_vek - opt_y_vek)
newclass_pres_z = norm(z_vek - opt_z_vek)

save metoda_newclass newclass_mi_count newclass_dual_medz newclass_prim_odch
newclass_dual_odch newclass_pres_x newclass_pres_y newclass_pres_z
pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% NEWTONOVE SMERY PRE LOGARITMICKU BARIEROVU METODU
function [dx, dy, dz] = log_newton_smer(A, x, z, mi)

X = diag(x);
Z = diag(z);
PS = mi * ones(length(x),1) - X * Z * ones(length(x),1);

% choleskeho rozklad
M = A * X * inv(Z) * A';
R = chol(M);
u = inv(R') * (-A * inv(Z) * PS);

dy = inv(R) * u;
dz = -A' * dy;
dx = inv(Z) * PS - inv(Z) * X * dz;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% METODA ZLATEHO REZU NA NAJDENIE OPTIMALNEJ DLZKY NEWTON
KROKU PRE LOGARITMICKU BARIEROVU METODU
function [alfa_opt, beta_opt] = log_newton_krok(c, b, x, dx, y, dy, z, dz, mi)

k=0; j=0;
poc_oper = 30;
x_podiel = 1000000;
z_podiel = 1000000;
for i=1:1:length(x)
   if dx(i)<0
      k=k+1;
      x_podiel(k) = -x(i) / dx(i);
   end
   if dz(i)<0
      j=j+1;
      z_podiel(j) = -z(i) / dz(i);
   end
end
alfa_max = min(x_podiel);
beta_max = min(z_podiel);

tau = (1 + sqrt(5)) / 2;
z_1 = 2 - tau;
z_2 = tau - 1;

% ALFA OPTIMALNE
```

```
for ii=1:1:poc_oper
    % pociatocne hodnoty alfa
    if ii==1
        alfa_min=0;
        alfa_1 = alfa_min + z_1 * (alfa_max - alfa_min);
        alfa_2 = alfa_min + z_2 * (alfa_max - alfa_min);
    end
    % funkcne hodnoty
    sum_1 = 0;
    sum_2 = 0;
    for i=1:1:length(x)
        sum_1 = sum_1 + log(x(i) + alfa_1 * dx(i));
        sum_2 = sum_2 + log(x(i) + alfa_2 * dx(i));
    end
    T_fun_1 = c'*(x + alfa_1 * dx) - mi * sum_1;
    T_fun_2 = c'*(x + alfa_2 * dx) - mi * sum_2;
    % nove hodnoty alfa
    if T_fun_1 >= T_fun_2
        alfa_min = alfa_1;
        alfa_1 = alfa_2;
        alfa_2 = alfa_min + z_2 * (alfa_max - alfa_min);
    end
    if T_fun_1 < T_fun_2
        alfa_max = alfa_2;
        alfa_2 = alfa_1;
        alfa_1 = alfa_min + z_1 * (alfa_max - alfa_min);
    end
end
alfa_opt = alfa_1;

% BETA OPTIMALNE
for ii=1:1:poc_oper
    % pociatocne hodnoty beta
    if ii==1
        beta_min = 0;
        beta_1 = beta_min + z_1 * (beta_max - beta_min);
        beta_2 = beta_min + z_2 * (beta_max - beta_min);
    end
    % funkcne hodnoty
    sum_1 = 0;
    sum_2 = 0;
    for i=1:1:length(x)
        sum_1 = sum_1 + log(z(i) + beta_1 * dz(i));
        sum_2 = sum_2 + log(z(i) + beta_2 * dz(i));
```

```
    end
    Q_fun_1 = -(b'*(y + beta_1 * dy) + mi * sum_1);
    Q_fun_2 = -(b'*(y + beta_2 * dy) + mi * sum_2);
    % nove hodnoty beta
    if Q_fun_1 >= Q_fun_2
        beta_min = beta_1;
        beta_1 = beta_2;
        beta_2 = beta_min + z_2 * (beta_max - beta_min);
    end
    if Q_fun_1 < Q_fun_2
        beta_max = beta_2;
        beta_2 = beta_1;
        beta_1 = beta_min + z_1 * (beta_max - beta_min);
    end
end
beta_opt = beta_1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NEWTONOVE SMERY PRE NELOGARITMICKU BARIEROVU METODU
function [dx, dy, dz] = nelog_newton_smer(A, x, z, mi, h, omega)

X = diag(x);
Z = diag(z);
PS_P = mi * X^h * ones(length(x),1) - X * Z * ones(length(x),1);
PS_D = (1 - h) * (mi^(1 - omega) * Z^omega * ones(length(x),1) -  X * Z * ones(length(x),1));

% choleskeho rozklad
M = A * X * inv(Z) * A';
R = chol(M);
u_P = inv(R') * (-A * inv(Z) * PS_P);
u_D = inv(R') * (-A * inv(Z) * PS_D);

dy_P = inv(R) * u_P;
dz_P = -A' * dy_P;
dx_P = (1 - omega) * (inv(Z) * PS_P - inv(Z) * X * dz_P);

dy_D = inv(R) * u_D;
dz_D = -A' * dy_D;
dx_D = (1 - omega) * (inv(Z) * PS_D - inv(Z) * X * dz_D);

dx = dx_P;
dy = dy_D;
```

```
dz = dz_D;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% METODA ZLATEHO REZU NA NAJDENIE OPTIMALNEJ DLZKY NEWTON KROKU PRE NELOGARITMICKU BARIEROVU METODU
function [alfa_opt, beta_opt] = nelog_newton_krok(c, b, x, dx, y, dy, z, dz, mi, h, omega)

k=0; j=0;
poc_oper = 30;
x_podiel = 1000000;
z_podiel = 1000000;
for i=1:1:length(x)
   if dx(i)<0
      k=k+1;
      x_podiel(k) = -x(i) / dx(i);
   end
   if dz(i)<0
      j=j+1;
      z_podiel(j) = -z(i) / dz(i);
   end
end
alfa_max = min(x_podiel);
beta_max = min(z_podiel);

tau = (1 + sqrt(5)) / 2;
z_1 = 2 - tau;
z_2 = tau - 1;

% ALFA OPTIMALNE
for ii=1:1:poc_oper
   % pociatocne hodnoty alfa
   if ii==1
      alfa_min=0;
      alfa_1 = alfa_min + z_1 * (alfa_max - alfa_min);
      alfa_2 = alfa_min + z_2 * (alfa_max - alfa_min);
   end
   % funkcne hodnoty
   sum_1 = 0;
   sum_2 = 0;
   for i=1:1:length(x)
      sum_1 = sum_1 + (x(i) + alfa_1 * dx(i))^h;
      sum_2 = sum_2 + (x(i) + alfa_2 * dx(i))^h;
   end
```

```
    T_fun_1 = c'*(x + alfa_1 * dx) - mi * (1/h) * sum_1;
    T_fun_2 = c'*(x + alfa_2 * dx) - mi * (1/h) * sum_2;
    % nove hodnoty alfa
    if T_fun_1 >= T_fun_2
        alfa_min = alfa_1;
        alfa_1 = alfa_2;
        alfa_2 = alfa_min + z_2 * (alfa_max - alfa_min);
    end
    if T_fun_1 < T_fun_2
        alfa_max = alfa_2;
        alfa_2 = alfa_1;
        alfa_1 = alfa_min + z_1 * (alfa_max - alfa_min);
    end
end
alfa_opt = alfa_1;

% BETA OPTIMALNE
for ii=1:1:poc_oper
    % pociatocne hodnoty beta
    if ii==1
        beta_min = 0;
        beta_1 = beta_min + z_1 * (beta_max - beta_min);
        beta_2 = beta_min + z_2 * (beta_max - beta_min);
    end
    % funkcne hodnoty
    sum_1 = 0;
    sum_2 = 0;
    for i=1:1:length(x)
        sum_1 = sum_1 + (z(i) + beta_1 * dz(i))^omega;
        sum_2 = sum_2 + (z(i) + beta_2 * dz(i))^omega;
    end
    Q_fun_1 = -(b'*(y + beta_1 * dy) + mi^(1 - omega) * (1/omega) * sum_1);
    Q_fun_2 = -(b'*(y + beta_2 * dy) + mi^(1 - omega) * (1/omega) * sum_2);
    % nove hodnoty beta
    if Q_fun_1 >= Q_fun_2
        beta_min = beta_1;
        beta_1 = beta_2;
        beta_2 = beta_min + z_2 * (beta_max - beta_min);
    end
    if Q_fun_1 < Q_fun_2
        beta_max = beta_2;
        beta_2 = beta_1;
        beta_1 = beta_min + z_1 * (beta_max - beta_min);
    end
```

```
end
beta_opt = beta_1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NEWTONOVE SMERY PRE NEW CLASS METODU
function [dx, dy, dz] = new_class_newton_smer(A, x, z, mi)

v = sqrt(x .* z / mi);
PS = 2 * (ones(length(v),1) - v);
A_new_1 = A * diag(x ./ v);
A_new_2 = A * diag(v ./ z);

delta_y = -inv(A_new_1 * A_new_2') * A_new_1 * PS;
delta_z = -A_new_2' * delta_y;
delta_x = PS - delta_z;

dy = delta_y;
dz = delta_z .* z ./ v;
dx = delta_x .* x ./ v;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% GENERATOR ULOH LP
clear all; clc;

ok=0;
while ok==0
   ok=1;
   m_rozmer = input('rozmer m (A je mxn) = ');
   n_rozmer = input('rozmer n (A je mxn) = ');
   if n_rozmer <= m_rozmer
      disp('n rozmer musi byt vacsi ako m rozmer')
      ok=0;
   end
end

% generovanie Ab a An
one_more_time=1;
while one_more_time==1
   one_more_time=0;
   for i=1:1:m_rozmer
      for ii=1:1:m_rozmer
         reg_A_mat(i,ii) = random('norm',10,12);
```

```
        end
    end
    if rank(reg_A_mat)~=m_rozmer || abs(det(reg_A_mat))<=1e-2
        one_more_time=1;
    end
end
reg_A_mat_inv = inv(reg_A_mat);
for i=1:1:m_rozmer
    for ii=1:1:(n_rozmer-m_rozmer)
        An_mat(i,ii) = random('norm',10,12);
    end
end
% generovanie qb > 0
bulhar_konst=0.01;
for i=1:1:m_rozmer
    qb_vek_help(i) = random('norm',10,12);
end
min_value = abs(min(qb_vek_help));
min_value = min_value + bulhar_konst * random('norm',10,4);
qb_vek = qb_vek_help + min_value;
qb_vek = qb_vek';
% vypocet p a qn
p_vek = -(reg_A_mat_inv' * qb_vek);
qn_vek = -An_mat' * p_vek;
% vypocet zn
zap_value=0;
opt_zn_vek = -qn_vek;
dlzka_opt_zn = length(opt_zn_vek);
for i=1:1:dlzka_opt_zn
    if opt_zn_vek(i) < 0
        zap_value=1;
        min_opt_zn_vek = min(opt_zn_vek);
    end
end
if zap_value==1
    opt_zn_vek = opt_zn_vek + abs(min_opt_zn_vek) + bulhar_konst *
random('norm',10,4);
else
    opt_zn_vek = opt_zn_vek + bulhar_konst * random('norm',10,4);
end
for i=1:1:length(opt_zn_vek)
    opt_zn_vek(i) = opt_zn_vek(i) + random('norm',6,2);
end
%vypocet zb
```

```
opt_zb_vek = zeros(m_rozmer,1);
%vypocet z opt a z start
opt_z_vek = [opt_zb_vek;opt_zn_vek];
start_z_vek = opt_z_vek + [qb_vek;qn_vek];
%vypocet y opt a y start
for i=1:1:m_rozmer
   opt_y_vek(i) = random('norm',10,12);
end
opt_y_vek = opt_y_vek';
start_y_vek  = opt_y_vek + p_vek;
% vypocet sn
zap_value=0;
for i=1:1:(n_rozmer - m_rozmer)
   sn_vek(i) =  random('norm',10,12);
   if sn_vek(i) < 0
      zap_value=1;
   end
end
sn_vek = sn_vek';
if zap_value==1
   sn_vek = sn_vek + abs(min(sn_vek)) + bulhar_konst * random('norm',10,4);
end
sb_vek = -inv(reg_A_mat) * (An_mat * sn_vek);
% vypocet x opt
zap_value=0;
opt_xb_vek = -sb_vek;
dlzka_opt_xb = length(opt_xb_vek);
for i=1:1:dlzka_opt_xb
   if opt_xb_vek(i) < 0
      zap_value=1;
   end
end
if zap_value==1
   min_opt_xb_vek = min(opt_xb_vek);
   opt_xb_vek = opt_xb_vek + abs(min_opt_xb_vek) + bulhar_konst * random('norm',10,4);
else
   opt_xb_vek = opt_xb_vek + bulhar_konst * random('norm',10,4);
end
for i=1:1:length(opt_xb_vek)
   opt_xb_vek(i) = opt_xb_vek(i) + random('norm',6,2);
end
opt_xn_vek = zeros((n_rozmer - m_rozmer),1);
opt_x_vek = [opt_xb_vek;opt_xn_vek];
```

```
% vypocet x start
start_x_vek = opt_x_vek + [sb_vek;sn_vek];
% vypocet pravych stran b, c
b_vek = reg_A_mat * opt_xb_vek;
cb_vek = reg_A_mat' * opt_y_vek;
cn_vek = An_mat' * opt_y_vek + opt_zn_vek;
A_mat = [reg_A_mat An_mat];
c_vek = [cb_vek;cn_vek];
% spolocna optimalna hodnota ucelovych funkcii
gama = c_vek' * opt_x_vek;

save priklad A_mat c_vek b_vek start_x_vek start_y_vek start_z_vek opt_x_vek opt_y_vek
opt_z_vek gama

pause
```