Univerzitet u Novom Sadu
Prirodno-matematički fakultet
Departman za matematiku i informatiku

Katarina Džepina

Gradijentne metode i metode konjugovanih gradijenata

– Master rad –

Novi Sad, 2020

Sadržaj

Uvod

Optimizacija predstavlja postupak nalaženja najboljeg rešenja nekog problema u određenom smislu i pod određenim uslovima. Po svojoj prirodi je veoma raznovrsna i ima široku primenu u svakodnevnom životu: u nauci, inženjerstvu, poslovnom upravljanju, vojnoj i kosmičkoj tehnologiji, porodici, ekonomskom sistemu i u mnogim drugim oblastima.

Iako optimizacija datira još od prvih problema pronalaženja ekstremuma, postaje samostalna oblast matematike tek od 1947. godine, kada je Dantcig (*Dancing*) predstavio dobro poznat simpleks algoritam za linearno programiranje.

Nakon 1950. godine, kada su metod konjugovanih gradijenata i kvazi-Njutnov metod predstavljeni, nelinearno programiranje se ubrzano razvija. Danas se različiti moderni metodi optimizacije mogu primeniti za rešavanje širokog spektra problema optimizacije i predstavljaju neophodno sredstvo za rešavanje problema u različitim oblastima.

Za razliku od zadataka linearnog programiranja, zadaci nelinearnog programiranja se ne mogu rešavati primenom nekog univerzalnog metoda (kao što je to simpleks metod za zadatke linearnog programiranja). Za zadatke nelinearnog programiranja je za svaki konkretan slučaj, u zavisnosti od njegovog matematičkog modela, dimenzija i karaktera nelinearnosti, potreban nov metod ili prilagođavanje nekog od postojećih metoda. U velikom broju slučajeva čak i ne postoji prikladni metod na osnovu kojeg se može naći optimalno rešenje formulisanog zadatka nelinearnog programiranja, što znači da postoji još uvek veliki broj nerešivih ili teško rešivih zadataka nelinearnog programiranja.

Mnogi od njih još uvek nisu rešivi jer ne postoje razvijeni algoritmi čija bi primena dala određene efekte. Primenljivost određenih algoritama procenjuje se na osnovu broja računskih operacija koje treba obaviti u procesu nalaženja rešenja. Neki algoritmi u određenim zadacima nelinearnog programiranja, čak i uz primenu savremenih računara, nisu uvek primenljivi.

Stoga, ovaj master rad se bavi gradijentnim metodama i metodama konjugovanih gradijenata kao dvema najbitnijim klasama metoda za rešavanje optimizacionih problema bez ograničenja. Ove klase metoda i dan danas služe kao osnova i motivacija za dalje razvijanje mnogih drugih algoritama čija bi primena dala korisne efekte na zadatake nelinearnog programiranja.

U prvom i drugom poglavlju dajemo osvrt na deo matematičke osnove koja je korišćena pri samoj izradi rada.

U trećem poglavlju opisujemo gradijentne metode koje su u osnovi metode linijskog pretraživanja. Uvešćemo pojam i uslovnih brojeva koji mogu da ukažu na veličinu broja iteracija, potrebnih da bi postigli unapred zadatu preciznost postupka. Da bismo rešili

lošu uslovljenost optimizacionog problema, u ovoj glavi ćemo se baviti i sa dijagonalnom skaliranjem gradijentnog pravca. U ovom delu rada razmotrićemo i Gaus-Njutnov metod i Vajsfelov metod (za rešavanje Ferma-Veberovog problema). Pomenuti metodi su specijalni slučajevi gradijentnog metoda. Na samom kraju ove glave data je teorijska analiza konvergencije, koja je jedan od najvažnijih aspekata svakog numeričkog postupka.

U četvrtom poglavlju ćemo razmotriti metode konjugovanih gradijenata, i to dva osnovna tipa: linearni i nelinearni. Linearni metod konjugovanih gradijenata nam služi za numeričko rešavanje linearnog sistema jednačina sa simetričnom pozitivno definitnom matricom, a koristićemo poznatu činjenicu da je ovaj problem ekvivalentan problemu minimizacije kvadratne funkcije koja poseduje matricu sa istim osobinama. Flečer i Rivs su, uz dve modifikacije, metod konjugovanih gradijenata primenili i na nelinearne funkcije. Ovaj njihov metod, kao i Polak-Ribierov metod i njegove modifikacije, takođe, razmatramo u ovoj glavi. Za pomenute metode konjugovanih gradijenata biće dati i teorijski rezultati vezani za konvergencije postupaka.

Poslednja, peta glava je zaključak. U okviru ove glave daćemo kratku analizu prethodno izloženih rezultata.

Veliku zahvalnost dugujem svom mentoru dr Goranu Radojevu, prvenstveno za razumevanje i posvećeno vreme, a potom za sve smernice, savete i sugestije tokom izrade rada.

Posebna zahvalnost prof. dr Sanji Rapajić kako na prenesenom znanju i zanimljivim predavanjima iz Operacionih istraživanja, tako i na nesebičnoj pomoći tokom izrade ovog rada, razumevanju i podršci.

Ovom prilikom bih se, takođe, zahvalila i prof. dr Zorani Lužanin na prenesenom znanju iz predmeta koja su obeležila moje master studije.

Takođe, zahvalila bih se svim profesorima i asistentima koji su bili deo mog studiranja. Svi su ostavili traga, i pomogli mi da bolje upoznam sebe kroz studiranje, koje za mene predstavlja bitan faktor u oblikovanju mladog čoveka.

Najveću zahvalnost i ljubav dugujem svom suprugu Vladimiru, mojim roditeljima, sestri Marini, bratu Marku i dragoj teta Gorici, kojima pripisujem najveću zaslugu za ono što jesam i za ono što sam postigla.

1 Pregled osnovnih osobina, definicija i teorema

Za uspešno praćenje ovog rada potrebno je, pre svega, da se upoznamo sa oznakama koje su korišćene pri izradi rada, kao i da damo uvid u osnovne definicije i teoreme. [1] [2] [5] [8] [11]

U čitavom radu za proizvoljno $n \in \mathbb{N}$, sa $\mathbb{R}^n$ označavamo n-dimenzionalni realni vektorski prostor, vektor kolona $\boldsymbol{x} = (\boldsymbol{x_1}, ..., \boldsymbol{x_n})^T$, gde je $\boldsymbol{x_i} \in \mathbb{R}$, $i = 1, 2, ..., n$, a za $m, n \in \mathbb{N}$ sa $\mathbb{R}^{m \times n}$ prostor svih $m \times n$ matrica sa realnim elementima. Pretežno će biti reč o kvadratnim matricama, te ćemo se ograničavati na prostor $\mathbb{R}^{n \times n}$.

Sa $\boldsymbol{x}^T$ označavamo transponovani vektor, a sa $\boldsymbol{A}^T$ transponovanu matricu. Sa $\boldsymbol{I}_n$ označavamo $n \times n$ jediničnu matricu, to jest matricu koja na dijagonali ima jedinice, a vandijagonalni elementi su nule. Kada je jasno o kojim dimenzijama se radi, pisaćemo samo $\boldsymbol{I}$. Ako drugačije nije naglašeno, sa $\boldsymbol{D} = diag(d_1, d_2, ..., d_n)$ ili nekad samo sa $\boldsymbol{D}$ označavamo $n \times n$ dijagonalnu matricu čiji su dijagonalni elementi redom $d_1, d_2, ..., d_n$, a vandijagonalni elementi nule.

U nastavku navodimo neke od definicija i teorema koje su nam neophodne za potpuno razumevanje rada.

1.1 Unutrašnji proizvod i norme

Definicija 1.1.1. (***Unutrašnji proizvod***) Unutrašnji proizvod na $\mathbb{R}^n$ je preslikavanje $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ sa sledećim osobinama:

- (***simetričnost***) $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$ za svako $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.
- (***aditivnost***) $\langle \boldsymbol{x}, \boldsymbol{y} + \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{x}, \boldsymbol{z} \rangle$ za svako $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n$.
- (***homogenost***) $\langle \lambda \boldsymbol{x}, \boldsymbol{y} \rangle = \lambda \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ za svako $\lambda \in \mathbb{R}$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.
- (***pozitivna definitnost***) $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geqslant 0$ za svako $\boldsymbol{x} \in \mathbb{R}^n$ i $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0$ ako i samo ako $\boldsymbol{x} = 0$.

Najčešći unutrašnji proizvod je takozvani skalarni proizvod definisan

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^T \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i, \text{ za svako } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

Kako je skalarni proizvod na neki način standarni unutrašnji proizvod, ako drugačije nije naglašeno, pretpostavljaćemo da je reč o standardnom.

Definicija 1.1.2. (***Norma vektora***) Norma $\|\cdot\|$ na $\mathbb{R}^n$ je funkcija $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ koja zadovoljava sledeće:

- (***nenegativnost***) $\|\boldsymbol{x}\| \geqslant 0$ za svako $\boldsymbol{x} \in \mathbb{R}^n$ i $\|\boldsymbol{x}\| = 0$ ako i samo ako $\boldsymbol{x} = 0$.
- (***pozitivna homogenost***) $\|\lambda\boldsymbol{x}\| = \lambda\|\boldsymbol{x}\|$ za svako $\boldsymbol{x} \in \mathbb{R}^n$ i $\lambda \in \mathbb{R}$.
- (***nejednakost trougla***) $\|\boldsymbol{x} + \boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ za svako $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

Prirodni način da se generiše norma na $\mathbb{R}^n$ jeste da se uzme unutrašnji proizvod $\langle \cdot, \cdot \rangle$ na $\mathbb{R}^n$ i definiše norma na sledeći način:

$$\|\boldsymbol{x}\| \equiv \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle} \text{ za svako } \boldsymbol{x} \in \mathbb{R}^n,$$

što se lako može proveriti po definiciji da je to zaista norma. Ako je unutrašnji proizvod skalarni, tada se norma naziva *Euklidska norma*:

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \text{ za svako } \boldsymbol{x} \in \mathbb{R}^n.$$

Standardna norma koju podrazumevamo na $\mathbb{R}^n$ jeste $\|\cdot\|_2$, te će indeks 2 često biti izostavljen u ovom radu.

Bitna nejednakost koja povezuje skalarni proizvod dva vektora sa njihovom normom je takozvana Koši-Švarcova nejednakost koju navodimo u nastavku, no pre toga dajemo definiciju zavisnosti vektora.

Definicija 1.1.3. (***Zavisnost vektora***) U vektorskom prostoru, niz vektora ($\boldsymbol{x_1}, \boldsymbol{x_2}$, ..., $\boldsymbol{x_n}$) je linearno zavisan, ako postoje skalari $\alpha_1, \alpha_2, ..., \alpha_n$, od kojih je bar jedan različit od nule, takvi da je

$$\alpha_1\boldsymbol{x_1} + \alpha_2\boldsymbol{x_2} + ... + \alpha_n\boldsymbol{x_n} = \boldsymbol{0}.$$

Niz vektora koji nije linearno zavisan, je linearno nezavisan.

Umesto niz vektora ($\boldsymbol{x_1}$,$\boldsymbol{x_2}$,...,$\boldsymbol{x_n}$) je linearno zavisan (nezavisan), govorićemo vektori ($\boldsymbol{x_1}$,$\boldsymbol{x_2}$,...,$\boldsymbol{x_n}$) su linearno zavisni (nezavisni).

Lema 1.1.1. (***Koši – Švarcova nejednakost***) *Za svako* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$,

$$|\boldsymbol{x}^T\boldsymbol{y}| \leqslant \|\boldsymbol{x}\|_2 \cdot \|\boldsymbol{y}\|_2.$$

Jednakost je zadovoljena ako i samo ako su vektori $\boldsymbol{x}, \boldsymbol{y}$ *linearno zavisni.*

Definicija 1.1.4. (***Matrična norma***) Norma $\|\cdot\|$ na $\mathbb{R}^{m\times n}$ je funkcija $\|\cdot\| : \mathbb{R}^{m\times n} \to \mathbb{R}$ koja zadovoljava sledeće:

- (***nenegativnost***) $\|\boldsymbol{A}\| \geqslant 0$ za svaku matricu $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, i $\|\boldsymbol{A}\| = 0$ ako i samo ako $\boldsymbol{A} = 0$.
- (***pozitivna homogenost***) $\|\lambda\boldsymbol{A}\| = |\lambda|\|\boldsymbol{A}\|$ za svaku matricu $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ i $\lambda \in \mathbb{R}$.
- (***nejednakost trougla***) $\|\boldsymbol{A} + \boldsymbol{B}\| \leqslant \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$ za svake matrice $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$.

1.2 Klasa pozitivno definitnih matrica

Da bismo mogli da okarakterišemo uslov optimalnosti drugog reda, koji se izražava preko matrice Hesijana, neophodno je da definišemo specijalnu klasu matrica, klasu pozitivno definitnih matrica.

Definicija 1.2.1. (***Pozitivna definitnost***).
1. Simetrična matrica $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ se naziva pozitivno semi-definitna, obeleženo sa $\boldsymbol{A} \succeq \boldsymbol{0}$, ako $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \geqslant 0$ za svako $\boldsymbol{x} \in \mathbb{R}^n$.
2. Simetrična matrica $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ se naziva pozitivno definitna, obeleženo sa $\boldsymbol{A} \succ \boldsymbol{0}$, ako $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0$ za svako $\boldsymbol{x} \in \mathbb{R}^n$.

U nastavku navodimo neke od bitnijih osobina klase pozitivno definitnih matrica.

Lema 1.2.1. *Neka je $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ pozitivno definitna matrica. Tada su dijagonalni elementi matrice $\boldsymbol{A}$ pozitivni.*

Lema 1.2.2. *Neka je $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ pozitivno semi-definitna matrica. Tada su dijagonalni elementi matrice $\boldsymbol{A}$ nenegativni.*

Teorema 1.2.1. (***Teorema o karakterizaciji karakterističnih korena***). *Neka je $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ simetrična matrica. Tada*

- *Matrica $\boldsymbol{A}$ je pozitivno definitna ako i samo ako su svi njeni karakteristični koreni pozitivni.*
- *Matrica $\boldsymbol{A}$ je pozitivno semi-definitna ako i samo ako su svi njeni karakteristični koreni nenegativni.*

Posledica 1.2.1. *Neka je $\boldsymbol{D} = diag(d_1, d_2, ..., d_n)$. Tada*

- *$\boldsymbol{D}$ je pozitivno definitna matrica ako su $d_i > 0$ za svako i.*
- *$\boldsymbol{D}$ je pozitivno semi-definitna matrica ako su $d_i \geqslant 0$ za svako i.*

Za svaku pozitivno definitnu matricu $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, može se definisati njen kvadratni koren $\boldsymbol{A}^{\frac{1}{2}}$ na sledeći način. Neka je $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\boldsymbol{T}}$ spektralna dekompozicija matrice $\boldsymbol{A}$, što znači da je $\boldsymbol{U}$ ortogonalna matrica i $\boldsymbol{D} = diag(d_1, ..., d_n)$ dijagonalna matrica čiji su dijagonalni elementi karakteristični koreni matrice $\boldsymbol{A}$. Kako je matrica A pozitivno definitna sledi da su $d_1, ..., d_n \geqslant 0$, i možemo definisati

$$\boldsymbol{A}^{\frac{1}{2}} = \boldsymbol{U}\boldsymbol{E}\boldsymbol{U}^{\boldsymbol{T}},$$

gde je $\boldsymbol{E} = diag(\sqrt{d_1}, ..., \sqrt{d_n})$. Očigledno je da važi

$$\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{A}^{\frac{1}{2}} = \boldsymbol{U}\boldsymbol{E}\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{U}\boldsymbol{E}\boldsymbol{U}^{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{E}\boldsymbol{E}\boldsymbol{U}^{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\boldsymbol{T}} = \boldsymbol{A}.$$

1.3 Karakteristični koreni i vektori

Neka je $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Nenula vektor $\boldsymbol{v} \in \mathbb{C}^n$ ($\mathbb{C}$ polje kompleksnih brojeva) se naziva *karakteristični vektor* matrice $\boldsymbol{A}$ ako postoji $\lambda \in \mathbb{C}$ za koje važi

$$\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}.$$

Skalar λ je karakteristični koren koji odgovara karakterističnom vektoru $\boldsymbol{v}$. U opštem slučaju, realne matrice mogu imati kompleksne sopstvene vrednosti, no poznato je da

svi karakteristični koreni simetričnih matrica su realni.

Karakteristične korene simetrične matrice $\boldsymbol{A}$ obeležavamo sa

$$\lambda_1(\boldsymbol{A}) \geqslant \lambda_2(\boldsymbol{A}) \geqslant \cdots \geqslant \lambda_n(\boldsymbol{A}).$$

Maksimalni karakteristični koren se takođe obeležava sa $\lambda_{max}(\boldsymbol{A})(= \lambda_1(\boldsymbol{A}))$, dok se minimalni karakteristični koren obeležava sa $\lambda_{min}(\boldsymbol{A})(= \lambda_n(\boldsymbol{A}))$. Napomenimo da se u ovom radu ograničavamo na simetrične realne matrice (za realnu matricu $\boldsymbol{A}$ kažemo da je simetrična ako važi $\boldsymbol{A^T} = \boldsymbol{A}$).

Postoji mnogo osobina koje daju vezu između pozitivno definitnih matrica i njihovih karakterističnih korena. Jedna od poznatijih je Kantorovičeva nejednakost čiji formalni oblik bez dokaza dajemo u nastavku.

Lema 1.3.1. (***Kantorovičeva nejednakost***). *Neka je $\boldsymbol{A}$ pozitivno definitna $n \times n$ matrica. Tada za svako $\boldsymbol{0} \neq \boldsymbol{x} \in \mathbb{R}^n$ važi nejednakost*

$$\frac{(\boldsymbol{x}^T\boldsymbol{x})^2}{(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x})(\boldsymbol{x}^T\boldsymbol{A}^{-1}\boldsymbol{x})^2} \geqslant \frac{4\lambda_{max}(\boldsymbol{A})\lambda_{min}(\boldsymbol{A})}{(\lambda_{max}(\boldsymbol{A}) + \lambda_{min}(\boldsymbol{A}))^2}\,.$$

1.4 Osnove topologije

Definicija 1.4.1. (***Otvorena i zatvorena lopta***). Otvorena lopta sa centrom $\boldsymbol{c} \in \mathbb{R}^n$ i radijusom r obeležavamo sa $B(\boldsymbol{c}, r)$ i definisano je sa

$$B(\boldsymbol{c}, r) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{c}\| < r\}.$$

Zatvorena lopta sa centrom u $\boldsymbol{c}$ i radijusom r obeležavamo sa $B[\boldsymbol{c}, r]$ i definisano je sa

$$B[\boldsymbol{c}, r] = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{c}\| \leqslant r\}.$$

Primetimo da norma koja se navodi u prethodnoj definiciji ne mora nužno da bude Euklidska norma. No, ako drugačije nije naglašeno, podrazumevamo upravo Euklidsku normu.

Definicija 1.4.2. (***Unutrašnje tačke***). Neka je dat skup $U \subseteq \mathbb{R}^n$. Tačka $\boldsymbol{c} \in U$ je unutrašnja tačka skupa U ako postoji $r > 0$ za koje $B(\boldsymbol{c}, r) \subseteq U$.

Skup svih unutrašnjih tačaka datog skupa U naziva se *unutrašnjost* skupa i obeležavamo sa $int(U)$:

$$int(U) = \{\boldsymbol{x} \in U : B(\boldsymbol{x}, r) \subseteq U \text{ za neko } r > 0\}.$$

Definicija 1.4.3. (***Otvoreni skupovi***). Otvoren skup je skup koji sadrži samo unutrašnje tačke. Drugim rečima, $U \subseteq \mathbb{R}^n$ je otvoren skup ako

$$\text{za svako } \boldsymbol{x} \in U \text{ postoji } r > 0 \text{ tako da } B(\boldsymbol{x}, r) \subseteq U.$$

1.5 Neprekidnost

Definicija 1.5.1. (***Neprekidnost funkcije***). Funkcija f definisana na skupu $S \subseteq \mathbb{R}^n$, neprekidna je u tački $\boldsymbol{x_0} \in S$ ako za svako $\varepsilon > 0$ postoji $\delta_\varepsilon > 0$ tako da za svako $\boldsymbol{x} \in S$

$$\|\boldsymbol{x} - \boldsymbol{x_0}\| < \delta_\varepsilon \implies |f(\boldsymbol{x}) - f(\boldsymbol{x_0})| < \varepsilon.$$

Funkcija je neprekidna na skupu, ako je neprekidna u svakoj tački tog skupa.

1.6 Diferencijabilnost

Neka je funkcija f definisana na skupu $S \subseteq \mathbb{R}^n$. Neka je $x \in int(S)$ i $\mathbf{0} \neq \boldsymbol{d} \in \mathbb{R}^n$. Ako limes

$$\lim_{t \to 0^+} \frac{f(\boldsymbol{x} + t\boldsymbol{d}) - f(\boldsymbol{x})}{t}$$

postoji, tada se on naziva izvod po pravcu funkcije f u tački $\boldsymbol{x}$ duž pravca vektora $\boldsymbol{d}$ i označavamo ga sa $f'(\boldsymbol{x}; \boldsymbol{d})$. Za svako $i = 1, 2, ..., n$ izvod po pravcu u $\boldsymbol{x}$ duž pravca $\boldsymbol{e_i}$ (vektor koji ima na i-tom mestu jedinicu, a svi ostali elementi su nule) naziva se *i-ti parcijalni izvod* i označavamo ga sa $\frac{\partial f}{\partial x_i}(\boldsymbol{x})$:

$$\frac{\partial f}{x_i}(\boldsymbol{x}) = \lim_{t \to 0^+} \frac{f(\boldsymbol{x} + t\boldsymbol{e_i}) - f(\boldsymbol{x})}{t}.$$

Ako svi parcijalni izvodi funkcije f postoje u tački $\boldsymbol{x} \in \mathbb{R}^n$, tada je gradijent funkcije f u tački $\boldsymbol{x}$ definisan sa vektorom kolona sačinjenog od svih parcijanih izvoda:

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial f}{x_1}(\boldsymbol{x}) \\ \frac{\partial f}{x_2}(\boldsymbol{x}) \\ \vdots \\ \frac{\partial f}{x_n}(\boldsymbol{x}) \end{pmatrix}.$$

Za funkciju f definisanu na otvorenom skupu $U \subseteq \mathbb{R}^n$ kažemo da je neprekidno diferencijabilna na celom U ako svi njeni parcijalni izvodi postoje i neprekidni su na U. Pod pretpostavkom neprekidne diferencijabilnosti imamo sledeću bitnu formulu za izvod po pravcu:

$$f'(\boldsymbol{x}; \boldsymbol{d}) = \nabla f^T(\boldsymbol{x})\boldsymbol{d}$$

za svako $\boldsymbol{x} \in U$ i $\boldsymbol{d} \in \mathbb{R}^n$.

Lema 1.6.1. *Neka je $f : U \to \mathbb{R}$ definisana na otvorenom skupu $U \subseteq \mathbb{R}^n$. Pretpostavimo da je f neprekidno diferencijabilna na U. Tada*

$$\lim_{\boldsymbol{d} \to 0} \frac{f(\boldsymbol{x} + \boldsymbol{d}) - f(\boldsymbol{x}) - \nabla f^T(\boldsymbol{x})\boldsymbol{d}}{\|\boldsymbol{d}\|} = 0 \text{ za svako } \boldsymbol{x} \in U.$$

Drugi način da se gornji rezultat zapiše jeste:

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla f^T(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}) + o(\|\boldsymbol{y} - \boldsymbol{x}\|),$$

gde je $o(\cdot) : \mathbb{R}^n_+ \to \mathbb{R}$ jednodimenzionalna funkcija koja zadovoljava $\frac{o(t)}{t} \to 0$ kada $t \to 0^+$.

Parcijalni izvodi $\frac{\partial f}{\partial x_i}$ su realne funkcije koje mogu da se parcijalno diferenciraju. Stoga, (i, j)-ti parcijalni izvodi funkcije f u tački $\boldsymbol{x} \in U$ (ako postoje) definisani su sa

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(\boldsymbol{x}) = \frac{\partial(\frac{\partial f}{\partial x_j})}{\partial x_i}(\boldsymbol{x}).$$

Za funkciju f definisanu na otvorenom skupu $U \subseteq \mathbb{R}^n$ kažemo da je *dva puta neprekidno diferencijabilna na U* ako svi parcijalni izvodi drugog reda postoje i neprekidni su na U. Pod pretpostavkom da je funkcija dva puta diferencijabilna, parcijalni izvodi drugog reda su simetrični, što znači da za svako $i \neq j$ i za svako $\boldsymbol{x} \in U$ važi

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(\boldsymbol{x}) = \frac{\partial^2 f}{\partial x_j \partial x_i}(\boldsymbol{x}).$$

Hesijan funkcije f u tački $\boldsymbol{x} \in U$ je matrica reda n:

$$\nabla^2 f(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial^2 x_2} & & \vdots \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial^2 x_n} \end{pmatrix},$$

pri tome su svi parcijalni izvodi drugog reda izračunati u tački $\boldsymbol{x}$. Kako je f dva puta neprekidno diferencijabilna na U, matrica Hesijana je simetrična. Postoje dva glavna aproksimativna rezultata (linearno i kvadratno) koja su direktna posledica Tejlorove aproksimativne teoreme, a koja ćemo kasnije u radu koristiti u nekim dokazima i objašnjenima.

Teorema 1.6.1. (***Teorema linearne aproksimacije***). *Neka je $f : U \to \mathbb{R}^n$ dva puta neprekidno diferencijabilna funkcija na otvorenom skupu $U \subseteq \mathbb{R}^n$, i neka $\boldsymbol{x} \in U$, $r > 0$ zadovoljava $B(\boldsymbol{x}, r) \in U$. Tada za svako $\boldsymbol{y} \in B(\boldsymbol{x}, r)$ postoji $\boldsymbol{\xi} \in [\boldsymbol{x}, \boldsymbol{y}]$ tako da*

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T (\boldsymbol{x} - \boldsymbol{y}) + \frac{1}{2} (\boldsymbol{y} - \boldsymbol{x})^T \nabla^2 f(\boldsymbol{\xi}) (\boldsymbol{y} - \boldsymbol{x}).$$

Teorema 1.6.2. (***Teorema kvadratne aproksimacije***). *Neka je $f : U \to \mathbb{R}^n$ dva puta neprekidno diferencijabilna funkcija na otvorenom skupu $U \subseteq \mathbb{R}^n$, i neka $\boldsymbol{x} \in U$, $r > 0$ zadovoljava $B(\boldsymbol{x}, r) \in U$. Tada za svako $\boldsymbol{y} \in B(\boldsymbol{x}, r)$*

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T (\boldsymbol{x} - \boldsymbol{y}) + \frac{1}{2} (\boldsymbol{y} - \boldsymbol{x})^T \nabla^2 f(\boldsymbol{x}) (\boldsymbol{y} - \boldsymbol{x}) + o(\|\boldsymbol{y} - \boldsymbol{x}\|^2).$$

1.7 Konveksnost i konkavnost

Definicija 1.7.1. (***Konveksnost funkcije***). Neka je $f : S \to \mathbb{R}$, $S \subseteq \mathbb{R}^n$. Funkcija f je konveksna na S ako za sve $\boldsymbol{x_1}, \boldsymbol{x_2} \in int(S)$ i sve $t \in [0, 1]$ važi nejednakost

$$f(t\boldsymbol{x_1} + (1 - t)\boldsymbol{x_2}) \leqslant t f(\boldsymbol{x_1}) + (1 - t) f(\boldsymbol{x_2}).$$

Ukoliko u prethodnoj definiciji važi suprotna nejednakost, tj. $\geqslant$, kažemo da je funkcija konkavna na S. U slučaju da u prethodnoj definiciji važi stroga nejednakost za sve $\boldsymbol{x_1}, \boldsymbol{x_2} \in int(S), \boldsymbol{x_1} \neq \boldsymbol{x_2}$ i sve $t \in (0, 1)$, kažemo da je funkcija f strogo konveksna na S.

U daljim razmatranjima ograničavamo se na konveksne funkcije, jer za konkavnu funkciju f, funkcija $(-f)$ je konveksna funkcija.

Teorema 1.7.1. *Neka je funkcija $f : S \to \mathbb{R}$, $S \subseteq \mathbb{R}^n$, dva puta diferencijabilna na S. Tada je funkcija f konveksna na S ako i samo ako $\nabla^2 f(\boldsymbol{x}) \geqslant \boldsymbol{0}$, za svako $\boldsymbol{x} \in int(S)$.*

2 Optimizacija bez ograničenja

Problem optimizacije bez ograničenja je problem oblika

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

gde je $f : \mathbb{R}^n \to \mathbb{R}$ funkcija cilja definisana na prostoru $\mathbb{R}^n$. Problem minimizacije se uvek može svesti na problem maksimizacije i obrnuto. Naime,

$$\max_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \min_{\boldsymbol{x}\in\mathbb{R}^n} -f(\boldsymbol{x}).$$

Relevantna literatura ovog poglavlja jeste [2], [12] i [20].

2.1 Globalni i lokalni optimum

Definicija 2.1.1. (***Optimalno rešenje***)**.** Dopustivo rešenje u kome funkcija cilja dostiže svoju optimalnu vrednost naziva se *optimalno rešenje* optimizacionog problema i označava se sa $\boldsymbol{x}^*$.

Definicija 2.1.2. (***Optimalna vrednost***)**.** Vrednost funkcije cilja koja odgovara optimalnom rešenju naziva se *optimalna vrednost* ili *optimum* i označava se sa $f^* = f(\boldsymbol{x}^*)$.

Iako ćemo posmatrati ceo prostor $\mathbb{R}^n$, u nastavku dajemo uopštene definicije globalnih i lokalnih ekstemuma.

Definicija 2.1.3. (***Globalni minimum i maksimum***)**.** Neka je $f : S \to \mathbb{R}$ definisana na skupu $S \subseteq \mathbb{R}^n$. Tada

- $\boldsymbol{x}^* \in S$ se naziva *tačka globalnog minimuma* funkcije f na S ako $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*)$ za svako $\boldsymbol{x} \in S$.
- $\boldsymbol{x}^* \in S$ se naziva *tačka striktnog globalnog minimuma* funkcije f na S ako $f(\boldsymbol{x}) > f(\boldsymbol{x}^*)$ za svako $\boldsymbol{x} \in S\backslash\{\boldsymbol{x}^*\}$.
- $\boldsymbol{x}^* \in S$ se naziva *tačka globalnog maksimuma* funkcije f na S ako $f(\boldsymbol{x}) \leqslant f(\boldsymbol{x}^*)$ za svako $\boldsymbol{x} \in S$.
- $\boldsymbol{x}^* \in S$ se naziva *tačka striktnog globalnog maksimuma* funkcije f na S ako $f(\boldsymbol{x}) < f(\boldsymbol{x}^*)$ za svako $\boldsymbol{x} \in S\backslash\{\boldsymbol{x}^*\}$.

Često ćemo umesto tačke globalnog minimuma govoriti tačka minimuma ili minimizator. Analogno i za tačku maksimuma, no o njoj će najmanje biti reči jer znamo da problem minimizacije može da se posmatra kao problem maksimizacije, a mi ćemo u ovom radu posmatrati problem optimizacije iz ugla minimizacije.

Dakle, vektor $\boldsymbol{x}^* \in S$ naziva se globalni optimum funkcije f na skupu S ako je ili globalni minimum ili globalni maksimum.

Skup svih globalnih minimizatora funkcije f na skupu S obeležavamo sa

$$\text{argmin}\{f(\boldsymbol{x}) : \boldsymbol{x} \in S\},$$

a skup svih globalnih maksimizatora od f na S sa

$$\text{argmax}\{f(\boldsymbol{x}) : \boldsymbol{x} \in S\}.$$

Definicija 2.1.4. (***Lokalni minimum i maksimum***)**.** Neka je $f : S \to \mathbb{R}$ definisana na skupu $S \subseteq \mathbb{R}^n$. Tada

- $\boldsymbol{x}^* \in S$ se naziva *tačka lokalnog minimuma* funkcije f na S ako postoji $r > 0$ za koje $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*)$ za svako $\boldsymbol{x} \in S \cap B(\boldsymbol{x}^*, r)$.
- $\boldsymbol{x}^* \in S$ se naziva *tačka striktnog lokalnog minimuma* funkcije f na S ako postoji $r > 0$ za koje $f(\boldsymbol{x}) > f(\boldsymbol{x}^*)$ za svako $\boldsymbol{x} \in S \cap B(\boldsymbol{x}^*, r) \backslash \{\boldsymbol{x}^*\}$.
- $\boldsymbol{x}^* \in S$ se naziva *tačka lokalnog maksimuma* funkcije f na S ako postoji $r > 0$ za koje $f(\boldsymbol{x}) \leqslant f(\boldsymbol{x}^*)$ za svako $\boldsymbol{x} \in S \cap B(\boldsymbol{x}^*, r)$.
- $\boldsymbol{x}^* \in S$ se naziva *tačka striktnog lokalnog maksimuma* funkcije f na S ako postoji $r > 0$ za koje $f(\boldsymbol{x}) < f(\boldsymbol{x}^*)$ za svako $\boldsymbol{x} \in S \cap B(\boldsymbol{x}^*, r) \backslash \{\boldsymbol{x}^*\}$.

Jasno je da je tačka globalnog minimuma (maksimuma) ujedno i tačka lokalnog minimuma (maksimuma), dok obrnuto ne mora da važi.

2.2 Uslov optimalnosti prvog reda

Dobro poznati rezultat kod jednodimenzionalne funkcije f definisane i diferencijabilne na intervalu (a, b) kaže da ako je tačka $x^* \in (a, b)$ lokalni maksimum ili minimum, tada je $f'(x^*) = 0$. Ovo svojstvo je poznato kao Fermaova teorema. Kada je u pitanju višedimenzionalni prostor, tada je u tačkama lokalnog optimuma gradijent jednak nuli. Takve uslove optimalnosti nazivamo *uslovi optimalnosti prvog reda*, jer se izražavaju preko izvoda prvog reda.

Teorema 2.2.1. (***Uslovi optimalnosti prvog reda***)**.** *Neka je $f : U \to \mathbb{R}$ funkcija definisana na $U \subseteq \mathbb{R}^n$. Pretpostavimo da je $\boldsymbol{x}^* \in int(U)$ tačka lokalnog optimuma i da svi parcijalni izvodi funkcije f postoje u $\boldsymbol{x}^*$. Tada je $\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$.*

Primetimo da navedena teorema predstavlja potreban uslov optimalnosti: gradijent nestaje u svim lokalnim tačkama optimuma koje su unutrašnje tačke domena funkcije f, dok obrnuto ne važi. Naime, postoje tačke koje nisu tačke lokalnog optimuma, ali je njihov gradijent jednak nuli. Jedan od takvih primera jeste jednodimenzionalna funkcija $f(x) = x^3$. Izvod takve funkcije je nula kada je $x = 0$, no ta tačka nije ni lokalni minimum ni lokalni maksimum, nego prevojna tačka. Kako tačke u kojima gradijent nestaje su jedini kandidati za lokalni optimum među svim tačkama unutrašnjosti domena funkcije f, zaslužuju i eksplicitnu definiciju koju navodimo u nastavku.

Definicija 2.2.1. (***Stacionarne tačke***)**.** Neka je $f : U \to \mathbb{R}$ funkcija definisana na $U \subseteq \mathbb{R}^n$. Pretpostavimo da je $\boldsymbol{x}^* \in int(U)$ i da je f diferencijabilna u nekoj okolini tačke $\boldsymbol{x}^*$. Tada je $\boldsymbol{x}^*$ stacionarna tačka funkcije f ako je $\nabla f(\boldsymbol{x}^*) = 0$.

Suštinski, Teorema 2.2.1. govori da su tačke lokalnih optimuma nužno i stacionarne tačke.

2.3 Uslov optimalnosti drugog reda

Teorema 2.3.1. (***Potrebni uslovi optimalnosti drugog reda***). *Neka je $f : U \to \mathbb{R}$ funkcija definisana na otvorenom skupu $U \subseteq \mathbb{R}$. Pretpostavimo da je f dva puta neprekidno diferencijabilna na U i neka je $\boldsymbol{x}^*$ stacionarna tačka. Tada važi sledeće:*

- *Ako je $\boldsymbol{x}^*$ tačka lokalnog minimuma funkcije f na U, tada je $\nabla^2 f(\boldsymbol{x}^*) \succeq \mathbf{0}$.*
- *Ako je $\boldsymbol{x}^*$ tačka lokalnog maksimuma funkcije f na U, tada je $\nabla^2 f(\boldsymbol{x}^*) \preceq \mathbf{0}$.*

Poslednji rezultat daje potreban uslov za lokalni optimum. Sledeća teorema nam govori o dovoljnom uslovu za striktni lokalni optimum.

Teorema 2.3.2. (***Dovoljan uslov optimalnosti drugog reda***). *Neka je $f : U \to \mathbb{R}$ funkcija definisana na otvorenom skupu $U \subseteq \mathbb{R}$. Pretpostavimo da je f dva puta neprekidno diferencijabilna na U i neka je $\boldsymbol{x}^*$ stacionarna tačka. Tada važi sledeće:*

- *Ako je $\nabla^2 f(\boldsymbol{x}^*) \succ \mathbf{0}$, tada je $\boldsymbol{x}^*$ tačka striktnog lokalnog minimuma funkcije f na U.*
- *Ako je $\nabla^2 f(\boldsymbol{x}^*) \prec \mathbf{0}$, tada je $\boldsymbol{x}^*$ tačka striktnog lokalnog maksimuma funkcije f na U.*

Primetimo da dovoljan uslov implicira jače svojstvo strikte lokalne optimalnosti, te pozitivna definitnost Hesijana nije neophodan uslov. Na primer, jednodimenzionalna funkcija $f(x) = x^4$ na $\mathbb{R}$ ima striktni lokalni minimum u $x = 0$, no $f''(0)$ nije pozitivno. Dakle, postoje stacionarne tačke koje nisu ni lokalni minimum ni lokalni maksimum i takve tačke nazivamo sedlaste tačke čiju definiciju dajemo u nastavku.

Definicija 2.3.1. (***Sedlasta tačka***). Neka je $f : U \to \mathbb{R}$ funkcija definisana na otvorenom skupu $U \subseteq \mathbb{R}^n$. Pretpostavimo da je f neprekidno diferencijabilna na U. Stacionarna tačka $\boldsymbol{x}^*$ se naziva *sedlasta tačka* funkcije f na U ako nije tačka ni lokalnog minimuma ni lokalnog maksimuma funkcije f na U.

2.4 Globalni uslovi optimalnosti

Naime, uslovi opisani u prethodnoj sekciji mogu da garantuju u najboljem slučaju lokalnu optimalnost stacionarnih tačaka jer se služe samo informacijama lokaliteta, odnosno sa vrednostima gradijenata i Hesijana u datim tačkama. Uslovi koji obezbeđuju globalnu konvergenciju moraju da koriste globalne informacije. Kao na primer da je Hesijan funkcije uvek pozitivno semi-definitan, da su sve stacionarne tačke takođe globalne tačke minimuma.

Teorema 2.4.1. *Neka je funkcija f dva puta neprekidno diferencijabilna i definisana na celom $\mathbb{R}^n$. Pretpostavimo da je $\nabla^2 f(\boldsymbol{x}) \succeq \mathbf{0}$ za svako $\boldsymbol{x} \in \mathbb{R}^n$. Neka je $\boldsymbol{x}^*$ stacionarna tačka funkcije f. Tada je $\boldsymbol{x}^*$ tačka globalnog minimuma funkcije f.*

2.5 Kvadratne funkcije

Kvadratne funkcije su bitna klasa funkcija koje se koriste u modeliranju mnogih optimizacionih problema, te ovu sekciju posvećujemo njima.

Definicija 2.5.1. Kvadratna funkcija na $\mathbb{R}^n$ je funkcija oblika

$$f(\boldsymbol{x}) = \boldsymbol{x}^T A \boldsymbol{x} + 2\boldsymbol{b}^T \boldsymbol{x} + c, \tag{2.1}$$

gde je $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ simetrična matrica, $\boldsymbol{b} \in \mathbb{R}^n$ i $c \in \mathbb{R}$.

Gradijent i Hesijan kvadratne funkcije imaju jednostavne analitičke formule:

$$\nabla f(\boldsymbol{x}) = 2\boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{b},$$
$$\nabla^2 f(\boldsymbol{x}) = 2\boldsymbol{A}.$$

U nastavku navodimo lemu bez dokaza sa nekoliko bitnih svojstava kvadratnih funkcija.

Lema 2.5.1. *Neka je* $f(\boldsymbol{x}) = \boldsymbol{x}^T A\boldsymbol{x} + 2\boldsymbol{b}^T\boldsymbol{x} + c$, *gde je* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *simetrična matrica,* $\boldsymbol{b} \in \mathbb{R}^n$ *i* $c \in \mathbb{R}$. *Tada*

- $\boldsymbol{x}$ *je stacionarna tačka funkcije* f *ako i samo ako* $\boldsymbol{A}\boldsymbol{x} = -\boldsymbol{b}$,
- *Ako je* $\boldsymbol{A} \succeq \boldsymbol{0}$, *tada je* $\boldsymbol{x}$ *tačka globalnog minimuma funkcije* f *ako i samo ako* $\boldsymbol{A}\boldsymbol{x} = -\boldsymbol{b}$,
- *Ako je* $\boldsymbol{A} \succ 0$, *tada je* $\boldsymbol{x} = -\boldsymbol{A}^{-1}\boldsymbol{b}$ *tačka striktnog globalnog minimuma funkcije* f.

2.6 Metode opadajućih pravaca

Posmatramo problem optimizacije bez ograničenja oblika

$$\min\{f(x) : x \in \mathbb{R}^n\}.$$

Pretpostavićemo da je zadata funkcija cilja $f : \mathbb{R}^n \to \mathbb{R}$ neprekidna i diferencijabilna na celom $\mathbb{R}^n$.

Metode bezuslovne optimizacije su uglavnom iterativnog tipa. Osnovna ideja je da se generiše niz tačaka $\boldsymbol{x_k} \in \mathbb{R}^n$, na sledeći način:

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + t_k\boldsymbol{d_k}, \quad k = 0, 1, 2...,$$

gde je $\boldsymbol{d_k}$ vektor koji određuje pravac kretanja iz tačke $\boldsymbol{x_k}$, a t_k realni parametar koji određuje dužinu koraka iz tačke $\boldsymbol{x_k}$ u pravcu vektora $\boldsymbol{d_k}$.

Niz vektora $\boldsymbol{d_k} \in \mathbb{R}^n$ i parametra $t_k \in \mathbb{R}$ obično biramo tako da niz vrednosti $f(\boldsymbol{x_k})$ monotono opada i konvergira nekom od minimuma funkcije cilja f u $\mathbb{R}^n$.

Definicija 2.6.1. Neka je $f : \mathbb{R}^n \to \mathbb{R}$ neprekidna i diferencijabilna funkcija na $\mathbb{R}^n$. Vektor $\boldsymbol{0} \neq \boldsymbol{d} \in \mathbb{R}^n$ se naziva ***opadajući pravac*** funkcije f u tački $\boldsymbol{x}$ ako je izvod po pravcu negativan, tj.

$$f'(\boldsymbol{x}; \boldsymbol{d}) = \nabla f(\boldsymbol{x})^T\boldsymbol{d} < 0.$$

Lema 2.6.1. *Neka je* f *neprekidno diferencijabilna na* $\mathbb{R}^n$, *i neka* $\boldsymbol{x} \in \mathbb{R}^n$. *Pretpostavimo da je* $\boldsymbol{d}$ *opadajući pravac funkcije* f *u tački* $\boldsymbol{x}$. *Tada postoji* $\varepsilon > 0$ *tako da*

$$f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x})$$

za svako $t \in (0, \varepsilon]$.

Dokaz. *Kako je* $f'(\boldsymbol{x}; \boldsymbol{d}) < 0$, *iz definicije izvoda po pravcu imamo*

$$\lim_{t\to 0^+} \frac{f(\boldsymbol{x} + t\boldsymbol{d}) - f(\boldsymbol{x})}{t} = f'(\boldsymbol{x}; \boldsymbol{d}) < 0$$

za svako $t \in (0, \varepsilon]$, *što implicira željeni rezultat.*

Algoritam* : *Metode opadajućih pravaca

***Korak* 0.** Izabrati početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$.
***Korak* 1.** Izabrati opadajući pravac $\boldsymbol{d_k}$.
***Korak* 2.** Naći dužinu koraka t_k koji zadovoljava $f(\boldsymbol{x_k} + t_k\boldsymbol{d_k}) < f(\boldsymbol{x_k})$.
***Korak* 3.** Odrediti narednu iteraciju $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + t_k\boldsymbol{d_k}$.
***Korak* 4.** Ako je izlazni kriterijum zadovoljen, tada zaustavljamo algoritam, i izlazni parametar je $\boldsymbol{x_{k+1}}$. U suprotnom, uzeti da je $k = k + 1$ i ići na Korak 1.

Naravno, sam ovakav šematski prikaz nam ne govori mnogo i njegova implementacija bez dodatnih detalja nije moguća. Prirodno nam se javljaju razna pitanja, kao na primer kakvu početnu tačku izabrati, šta je izlazni kriterijum algoritma i slično. Glavna razlika između različitih metoda jeste upravo izbor pravca i veličine koraka, gde u radu razmatramo neke od mogućnosti. Najčešći izlazni kriterijum koji se nalazi u literaturi i praksi jeste $\|\nabla f(\boldsymbol{x_{k+1}})\| \leqslant \varepsilon$, koji koristimo u ovom poglavlju.

Proces traženja koraka t naziva se linijsko pretraživanje jer za cilj ima da minimizira jednodimenzionalnu funkciju $g(t) = f(\boldsymbol{x_k} + t\boldsymbol{d_k})$. Postoji više načina za traženje koraka, no mi navodimo tri najpoznatija:

- ***konstantan korak* :** $t_k = t$ za svako k
- ***tačno linijsko pretraživanje* :** t_k je minimizator funkcije f duž pravca $\boldsymbol{x_k} + t\boldsymbol{d_k}$:
$$t_k \in \underset{t \geqslant 0}{\operatorname{argmin}}\, f(\boldsymbol{x_k} + t\boldsymbol{d_k}).$$
- ***linijsko pretraživanje unazad* :** Metoda koja zahteva tri parametra $s > 0$, $\alpha \in (0, 1)$, $\beta \in (0, 1)$. Prvo uzimamo da je t_k jednak inicijalnom s. Zatim, dok
$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_k} + t_k\boldsymbol{d_k}) < -\alpha t_k \nabla f(\boldsymbol{x_k})^T d_k,$$
uzimamo za t_k da je jednako βt_k. Drugim rečima, korak je izabran tako da je $t_k = s\beta^{i_k}$, pri tome, i_k je najmanji nenegativan celi broj za koje je uslov
$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_k} + s\beta^{i_k}\boldsymbol{d_k}) \geqslant -\alpha s\beta^{i_k} \nabla f(\boldsymbol{x_k})^T \boldsymbol{d_k}$$
zadovoljen.

Glavna prednost konstantnog koraka jeste njegova jednostavnost iako je iz ovog aspekta nejasno kako ga biramo. Tačno linijsko pretraživanje ima manu što je nekada nemoguće naći minimizator. Iako treća opcija ne daje tačno linijsko pretraživanje, daje nam dovoljno dobar korak, pod tim podrazumevamo da je zadovoljen uslov dovoljnog pada, to jest:
$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_k} + t_k\boldsymbol{d_k}) \geqslant -\alpha t_k \nabla f(\boldsymbol{x_k})^T \boldsymbol{d_k}. \tag{2.2}$$

U nastavku dajemo lemu koja tvrdi da je uslov dovoljnog pada uvek zadovoljen za dovoljno malo t_k.

Lema 2.6.2. *Neka je f neprekidno diferencijabilna funkcija na celom $\mathbb{R}^n$, i neka je $\boldsymbol{x} \in \mathbb{R}^n$. Pretpostavimo da je $\mathbf{0} \neq \boldsymbol{d} \in \mathbb{R}^n$ opadajući pravac funkcije f u tački $\boldsymbol{x}$ i neka $\alpha \in (0, 1)$. Tada postoji $\varepsilon > 0$ tako da nejednakost*
$$f(\boldsymbol{x}) - f(\boldsymbol{x} + t\boldsymbol{d}) \geqslant -\alpha t \nabla f(\boldsymbol{x})^T \boldsymbol{d}$$
važi za svako $t \in [0, \varepsilon]$.

Dokaz. *Kako je f neprekidno diferencijabilna sledi*

$$f(\boldsymbol{x}+t\boldsymbol{d}) = f(\boldsymbol{x}) + t\nabla f(\boldsymbol{x})^T\boldsymbol{d} + o(t\|\boldsymbol{d}\|),$$

tako da

$$f(\boldsymbol{x}) - f(\boldsymbol{x}+t\boldsymbol{d}) = -\alpha t\nabla f(\boldsymbol{x})^T\boldsymbol{d} - (1-\alpha)t\nabla f(\boldsymbol{x})^T\boldsymbol{d} - o(t\|\boldsymbol{d}\|). \tag{2.3}$$

Pošto je $\boldsymbol{d}$ opadajući pravac funkcije f u tački $\boldsymbol{x}$ imamo

$$\lim_{t\to 0^+} \frac{(1-\alpha)t\nabla f(\boldsymbol{x})^T\boldsymbol{d} + 0(t\|\boldsymbol{d}\|)}{t} = (1-\alpha)\nabla f(\boldsymbol{x})^T\boldsymbol{d} < 0.$$

Stoga, postoji $\varepsilon > 0$ tako da za svako $t \in (0, \varepsilon]$ nejednakost

$$(1-\alpha)t\nabla f(\boldsymbol{x})^T\boldsymbol{d} + o(t\|\boldsymbol{d}\|) < 0$$

važi, što u kombinaciji sa (2.3) daje željeni rezultat.

Primer 2.6.1. *(Tačno linijsko pretraživanje za kvadratnu funkciju). Neka je $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{b}^T\boldsymbol{x} + c$, gde je $\boldsymbol{A}$ $n\times n$ pozitivno definitna matrica, $\boldsymbol{b}\in\mathbb{R}^n$, i $c\in\mathbb{R}$. Neka je $\boldsymbol{x}\in\mathbb{R}^n$ i neka je $\boldsymbol{d}\in\mathbb{R}^n$ opadajući pravac funkcije f u tački $\boldsymbol{x}$. Želimo da nađemo eksplicitnu formulu koraka generisanog sa tačnim linijskim pretraživanjem, a to je rešenje problema*

$$\min_{t\geqslant 0} f(\boldsymbol{x}+t\boldsymbol{d}).$$

Definišimo prethodnu funkciju sa $g(t)$. Ako je raspišemo imamo:

$$\begin{aligned} g(t) = f(\boldsymbol{x}+t\boldsymbol{d}) &= (\boldsymbol{x}+t\boldsymbol{d})^T\boldsymbol{A}(\boldsymbol{x}+t\boldsymbol{d}) + 2\boldsymbol{b}^T(\boldsymbol{x}+t\boldsymbol{d}) + c \\ &= (\boldsymbol{d}^T\boldsymbol{A}\boldsymbol{d})t^2 + 2(\boldsymbol{d}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{b})t + \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{b}^T\boldsymbol{x} + c \\ &= (\boldsymbol{d}^T\boldsymbol{A}\boldsymbol{d})t^2 + 2(\boldsymbol{d}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{b})t + f(\boldsymbol{x}) \end{aligned}$$

Kako je $g'(t) = 2(\boldsymbol{d}^T\boldsymbol{A}\boldsymbol{d})t + 2\boldsymbol{d}^T(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})$ i $\nabla f(\boldsymbol{x}) = 2(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})$, sledi da je

$$\begin{aligned} g'(t) = 0 &\iff 2(\boldsymbol{d}^T\boldsymbol{A}\boldsymbol{d})t + 2\boldsymbol{d}^T(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}) = 0 \\ &\iff 2(\boldsymbol{d}^T\boldsymbol{A}\boldsymbol{d})t = -2\boldsymbol{d}^T(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}) \end{aligned}$$

odakle zaključujemo da je

$$t = \bar{t} \equiv -\frac{\boldsymbol{d}^T\nabla f(\boldsymbol{x})}{2\boldsymbol{d}^T\boldsymbol{A}\boldsymbol{d}}.$$

Pošto je $\boldsymbol{d}$ opadajući pravac funkcije f u tački $\boldsymbol{x}$, sledi $\boldsymbol{d}^T\nabla f(\boldsymbol{x}) < 0$, te je $\bar{t} > 0$. Kako je drugi izvod funkcije g uvek pozitivan, što se lako može proveriti, zaključujemo da je $\bar{t}$ dobijeno tačnim linijskim pretraživanjem.

□

3 Gradijentne metode

Postupci za rešavanje problema optimizacija bez ograničenja dele se na gradijentne i negradijentne metode. Negradijentne metode koriste samo vrednosti funkcija, a ne i njene izvode. Gradijentne metode koriste izvode funkcije, i dele se na metode prvog i drugog reda.

Metode prvog reda koriste samo prvi izvod funkcije cilja. Najpoznatiji među gradijentnim metodama prvog reda jeste Košijev metod najbržeg pada, koji datira iz 1847. godine. Sam metod jeste dobar, jer se dolazi do optimalnog rešenja iz bilo koje početne tačke, čime se postiže globalna konvergencija ali je relativno spora, jer linearno konvergira. Otuda i potreba za razvijanjem modifikovanih i sličnih metoda radi brže konvergencije.

U ovom master radu ćemo se baviti iskljućimo gradijentim metodama prvog reda, [2], [6].

3.1 Metod najbržeg pada

Metod najbržeg pada je jedan od osnovnih metoda za rešavanje problema minimizacije bez ograničenja. Pošto koristi pravac negativnog gradijenta kao opadajući pravac, u literaturi se ovaj metod često naziva gradijentni metod. Dakle, kod metode najbržeg pada je $\boldsymbol{d_k} = -\nabla f(\boldsymbol{x_k})$. Da je to zaista opadajući pravac, kad god je $\nabla f(\boldsymbol{x_k}) \neq \mathbf{0}$, pokazujemo na sledeći način:

$$f'(\boldsymbol{x_k}; -\nabla f(\boldsymbol{x_k})) = -\nabla f(\boldsymbol{x_k})^T \nabla f(\boldsymbol{x_k}) = -\|\nabla f(\boldsymbol{x_k})\|^2 < 0.$$

Pored toga, kod opadajućih pravaca negativni gradijent predstavlja i metod najbržeg pada, što znaci da normalizovan nagib $-\nabla f(\boldsymbol{x_k})/\|f(\boldsymbol{x_k})\|$ odgovara minimalnom izvodu po pravcu medju svim normalizovanim pravcima. U lemi koja sledi dajemo formalan dokaz.

Lema 3.1.1. *Neka je f neprekidno diferencijabilna funkcija, i neka je $\boldsymbol{x} \in \mathbb{R}^n$ nestacionarna tačka ($\nabla f(\boldsymbol{x}) \neq \mathbf{0}$). Optimalno rešenje problema*

$$\min_{\boldsymbol{d} \in \mathbb{R}^n} \{f'(\boldsymbol{x}; \boldsymbol{d}) : \|\boldsymbol{d}\| = 1\} \tag{3.1}$$

je $\boldsymbol{d} = -\frac{\nabla f(\boldsymbol{x})}{\|\nabla f(\boldsymbol{x})\|}$.

Dokaz. *Kako je $f'(\boldsymbol{x}; \boldsymbol{d}) = \nabla f(\boldsymbol{x})^T \boldsymbol{d}$, problem (3.1) ekvivalentan je*

$$\min_{\boldsymbol{d} \in \mathbb{R}^n} \{\nabla f(\boldsymbol{x})^T \boldsymbol{d} : \|\boldsymbol{d}\| = 1\}.$$

Primenom Koši-Švarc nejednakosti i koristeći $\|\boldsymbol{d}\| = 1$, imamo

$$\nabla f(\boldsymbol{x})^T \boldsymbol{d} \geqslant -\|\nabla f(\boldsymbol{x})\| \|\boldsymbol{d}\| = -\|\nabla f(\boldsymbol{x})\|.$$

Prema tome, $-\|\nabla f(\boldsymbol{x})\|$ *je donja granica optimalne vrednosti problema (3.1). S druge strane, ako uzmemo da je* $\boldsymbol{d} = -\frac{\nabla f(\boldsymbol{x})}{\|\nabla f(\boldsymbol{x})\|}$ *i zamenimo u funkciji cilja (3.1) dobijamo*

$$f'\Big(\boldsymbol{x}, -\frac{\nabla f(\boldsymbol{x})}{\|\nabla f(\boldsymbol{x})\|}\Big) = -\nabla f(\boldsymbol{x})^T \frac{\nabla f(\boldsymbol{x})}{\|\nabla f(\boldsymbol{x})\|} = -\|\nabla f(\boldsymbol{x})\|,$$

i tako dolazimo do zaključka da je donja granica $-\|\nabla f(\boldsymbol{x})\|$ *postignuta u* $\boldsymbol{d} = -\frac{\nabla f(\boldsymbol{x})}{\|\nabla f(\boldsymbol{x})\|}$, *što implicira da je to optimalno rešenje našeg problema (3.1).*

U nastavku predstavljamo šematski prikaz metode najbržeg pada sa standardnim izlaznim kriterijumom.

Algoritam : Metod najbržeg pada

***Korak* 0.** Definisati ulazni parametar tolerancije, ε.
***Korak* 1.** Izabrati početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$.
***Korak* 2.** Izabrati korak t_k metodom linijskog pretraživanja funkcije

$$g(t) = f(\boldsymbol{x_k} - t\nabla f(\boldsymbol{x_k})).$$

***Korak* 3.** Odrediti narednu iteraciju $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - t_k \nabla f(\boldsymbol{x_k})$.
***Korak* 4.** Ako je $\|\nabla f(\boldsymbol{x_{k+1}})\| < \varepsilon$, tada zaustavljamo algoritam, izlazni parametar je $\boldsymbol{x_{k+1}}$. U suprotnom, uzeti da je $k = k + 1$ i ići na Korak 2.

Primer 3.1.1. *(Primena metode najbržeg pada sa tačnim linijskim pretraživanjem na kvadratnu funkciju). Posmatramo trodimenzionalni problem minimizacije*

$$\min_{x,y,z} x^2 + 2y^2 + 4z^2 \tag{3.2}$$

čije je optimalno rešenje $(x, y, z) = (0, 0, 0)$ *sa optimalnom vrednošću 0. Konstruišemo funkciju u MATLAB-u pod nazivom* `gmk`*, koja nalazi do praga odredjene tolerancije optimalna rešenja problema minimizacije oblika*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \{\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{b}^T \boldsymbol{x}\},$$

gde je $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *pozitivno definitna matrica i* $\boldsymbol{b} \in \mathbb{R}^n$. *Kao što smo videli u prethodnom teorijskom primeru, za tačno linijsko pretraživanje u svakoj iteraciji* k, *korak* t_k *ima eksplicitni oblik* $\dfrac{\|\nabla f(\boldsymbol{x_k})\|^2}{2\nabla f(\boldsymbol{x_k})^T \boldsymbol{A} \nabla f(\boldsymbol{x_k})}$ *što primenjujemo u MATLAB funkciji koju opisujemo u nastavku.*

```
function [x,fun_vr]=gmk(A,b,x0,epsilon)
% INPUT
% ======================
% A ....... pozitivno definitna matrica funckije cilja
% b ....... vektor kolone linearnog dela problema optimizacije
% x0 ...... inicijalna tacka metoda
% epsilon . parametar tolerancije
% OUTPUT
% ======================
% x ....... optimalno resenje problema min(x^T A x+2 b^T x)
% fun_vr .. optimalna vrednost funkcije uz prag tolerancije
```

```
x=x0;
iter=0;
grad=2*(A*x+b);

while (norm(grad) > epsilon) && (iter < 100)
    iter=iter+1;
    t=norm(grad)^2/(2*grad'*A*grad);
    x=x-t*grad;
    grad=2*(A*x+b);
    fun_vr=x'*A*x+2*b'*x;
    fprintf('broj_iter = %3d norma_grad = %2.6f fun_vr = %e \n',...
        iter,norm(grad),fun_vr);
end
```

Naime, da bismo rešili naš problem primenom MATLAB funkcije, za prag tolerancije biramo $\varepsilon = 10^{-5}$ i za početni vektor $x_0 = (2, 1, 1)^T$. Matrica $\boldsymbol{A}$ kao i vektor $\boldsymbol{b}$, poznati su nam iz našeg problema koji posmatramo. Dakle, izvršavamo sledeću MATLAB komandu:
`»[x,fun_vr]=gmk([1,0,0;0,2,0;0,0,4],[0;0;0],[2;1;1],1e-5)`

Izlaz navedene komande je:

```
broj_iter =  1 norma_grad = 3.754222 fun_vr = 2.421053e+00
broj_iter =  2 norma_grad = 2.387852 fun_vr = 7.411613e-01
broj_iter =  3 norma_grad = 1.177492 fun_vr = 2.528891e-01
broj_iter =  4 norma_grad = 0.814752 fun_vr = 8.635091e-02
broj_iter =  5 norma_grad = 0.402082 fun_vr = 2.949148e-02
broj_iter =  6 norma_grad = 0.278263 fun_vr = 1.007225e-02
broj_iter =  7 norma_grad = 0.137323 fun_vr = 3.439988e-03
broj_iter =  8 norma_grad = 0.095035 fun_vr = 1.174863e-03
broj_iter =  9 norma_grad = 0.046900 fun_vr = 4.012521e-04
broj_iter = 10 norma_grad = 0.032458 fun_vr = 1.370401e-04
broj_iter = 11 norma_grad = 0.016018 fun_vr = 4.680343e-05
broj_iter = 12 norma_grad = 0.011085 fun_vr = 1.598483e-05
broj_iter = 13 norma_grad = 0.005471 fun_vr = 5.459315e-06
broj_iter = 14 norma_grad = 0.003786 fun_vr = 1.864526e-06
broj_iter = 15 norma_grad = 0.001868 fun_vr = 6.367934e-07
broj_iter = 16 norma_grad = 0.001293 fun_vr = 2.174847e-07
broj_iter = 17 norma_grad = 0.000638 fun_vr = 7.427779e-08
broj_iter = 18 norma_grad = 0.000442 fun_vr = 2.536817e-08
broj_iter = 19 norma_grad = 0.000218 fun_vr = 8.664019e-09
broj_iter = 20 norma_grad = 0.000151 fun_vr = 2.959032e-09
broj_iter = 21 norma_grad = 0.000074 fun_vr = 1.010601e-09
broj_iter = 22 norma_grad = 0.000052 fun_vr = 3.451517e-10
broj_iter = 23 norma_grad = 0.000025 fun_vr = 1.178800e-10
broj_iter = 24 norma_grad = 0.000018 fun_vr = 4.025969e-11
broj_iter = 25 norma_grad = 0.000009 fun_vr = 1.374993e-11
```

Primetimo da se metod zaustavio nakon 25 iteracija sa rešenjem koji je blizu optimalnog:

```
x =
   1.0e-05 *
   0.3472
   0.0000
   -0.0652
```

u kojem je vrednost funkcije:

```
fun_vr =
   1.3750e-11
```

Poznato je da su pravci metode najbržeg pada sa tačnim linijskim pretraživanjem međusobno ortogonalni. Formalni oblik ove osobine dajemo u lemi koja sledi.

Lema 3.1.2. *Neka je* $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ *niz generisan metodom najbržeg pada sa tačnim linijskim pretraživanjem za rešavanje problema minimizacije neprekidno diferencijabilne funkcije* f*. Tada za svako* $k = 0, 1, 2...$ *važi*

$$(\boldsymbol{x_{k+2}} - \boldsymbol{x_{k+1}})^T(\boldsymbol{x_{k+1}} - \boldsymbol{x_k}) = 0.$$

Dokaz. *Iz definicije metoda imamo* $\boldsymbol{x_{k+2}} - \boldsymbol{x_{k+1}} = -t_{k+1}\nabla f(\boldsymbol{x_{k+1}})$ *i* $\boldsymbol{x_{k+1}} - \boldsymbol{x_k} = -t_k\nabla f(\boldsymbol{x_k})$*. Dakle, treba da dokažemo da je* $\nabla f(\boldsymbol{x_k})^T\nabla f(\boldsymbol{x_{k+1}}) = 0$*.*
Kako je

$$t_k \in \operatorname*{argmin}_{t\geqslant 0}\{g(t) \equiv f(\boldsymbol{x_k} - t\nabla f(\boldsymbol{x_k}))\},$$

i optimalno rešenje nije $t_k = 0$*, sledi da je* $g'(t_k) = 0$*. Stoga,*

$$-\nabla f(\boldsymbol{x_k})^T\nabla f(\boldsymbol{x_k} - t_k\nabla f(\boldsymbol{x_k})) = 0,$$

odakle dobijamo $\nabla f(\boldsymbol{x_k})^T\nabla f(\boldsymbol{x_{k+1}}) = 0$ *što je trebalo pokazati.*

Primer 3.1.2. *(Primena metode najbržeg pada sa konstantnim korakom). Posmatramo isti problem optimizacije kao i u prethodnom primeru,*

$$\min_{x,y,z} x^2 + 2y^2 + 4z^2.$$

Da bismo rešili ovaj problem, konstruišemo sledeću MATLAB funkciju koja primenjuje metod najbržeg pada sa konstantnim korakom na proizvoljne funkcije cilja.

```
function [x, fun_vr] = gm_konst(f,g,x0,t,epsilon)
%gradijentni metod sa konstantnim korakom
%
%INPUT
%====================
%f.........funkcija cilja
%g.........gradijent funkcije cilja
%x0........pocetna tacka
%t.........korak
%epsilon...prag tolerancije
%OUTPUT
%====================
%x.........optimalno resenje funkcije minf(x) uz prag tolerancije epsilon
%fun_vr...optimalna vrednost funkcije cilja
x=x0;
grad=g(x);
iter=0;
while (norm(grad) > epsilon) && (iter < 100)
    iter=iter+1;
    x=x-t*grad;
    fun_vr=f(x);
    grad=g(x);
    fprintf('iter_number = %3d norm_grad = %2.6f fun_vr = %e \n',...
        iter,norm(grad),fun_vr);
end
```

Možemo primeniti metod najbržeg pada sa korakom $t_k = 0.1$ i inicijalnim vektorom $x_0 = (2, 1, 1)^T$ izvršavanjem MATLAB funkcije:

```
» A=[1,0,0;0,2,0;0,0,4];
» [x,fun_vr]=gm_konst(@(x)x'*A*x,@(x)2*A*x,[2;1;1],0.1,1e-05)
```

Pri tome dobijamo:

broj_iter = 1 norma_grad = 4.308132 fun_vr = 3.440000e+00
broj_iter = 2 norma_grad = 2.954590 fun_vr = 1.904000e+00
broj_iter = 3 norma_grad = 2.223712 fun_vr = 1.142144e+00
broj_iter = 4 norma_grad = 1.718504 fun_vr = 7.046912e-01
: :
broj_iter = 55 norma_grad = 0.000019 fun_vr = 8.749003e-11
broj_iter = 56 norma_grad = 0.000015 fun_vr = 5.599362e-11
broj_iter = 57 norma_grad = 0.000012 fun_vr = 3.583592e-11
broj_iter = 58 norma_grad = 0.000010 fun_vr = 2.293499e-11

Metod se zaustavio nakon 58 iteracija sa približnim optimalnim rešenjem:

```
x =
   1.0e-05 *
   0.4789
   0.0000
   0.0000
```

u kojem je vrednost funkcije:

```
fun_vr =
   2.2935e-11
```

Očekivan je velik broj iteracija jer je izabrani korak previše mali, dok u slučaju izbora znatno većeg konstantnog koraka može da dođe do divergencije metoda. Na primer, uzmajući da je konstantan korak 100, dolazi do divergencije, što vidimo u nastavku.

```
» [x,fun_vr]=gm_konst(@(x)x'*A*x,@(x)2*A*x,[2;1;1],100,1.e-5)
```

broj_iter = 1 norma_grad = 6636.150691 fun_vr = 3.030410e+06
broj_iter = 2 norma_grad = 5149192.627674 fun_vr = 1.687186e+12
: :
broj_iter = 100 norma_grad = Inf fun_vr = Inf

Napomena: Inf je skraćenica od 'Infinity' u softverskom paketu MATLAB i nagoveštava broj iz beskonačnosti.

Primetimo da u ovom slučaju dobijamo divergenciju metoda, ali isto tako da se naš algoritam završava nakom 100 iteracija zbog dodatnog izlaznog kriterijuma, a to je da smo pored standardnog izlaznog kriterijuma $\|\nabla f(\boldsymbol{x_{k+1}})\| \leqslant \boldsymbol{\varepsilon}$ ograničili broj iteracija. Ako iz koda isključimo dodatni izlazni kriterijum koji se odnosi na maksimalan broj iteracija koji dozvoljavamo i ponovo izvršimo algoritam, dobijamo sledeće:

broj_iter = 1 norma_grad = 6636.150691 fun_vr = 3.030410e+06
broj_iter = 2 norma_grad = 5149192.627674 fun_vr = 1.687186e+12
: :
broj_iter = 107 norma_grad = NaN fun_vr = NaN

Napomena: NaN je skraćenica od 'Not a Number' u softveru MATLAB i nagoveštava divergenciju.

Dakle, naš primer zaista divergira, što se vidi po 107. iteraciji. □

Naime, postavlja se pitanje kako da izaberemo konstantan korak da ne bude previše velik da bi izbegli divergenciju, a ni previše mali da ne bi imali jako sporu konvergenciju. Nešto kasnije u radu posmatraćemo ovaj problem iz teoretskog aspekta.

Primer 3.1.3. *(Primena metode najbržeg pada sa linijskim pretraživanjem unazad). I dalje posmatramo isti problem optimizacije*

$$\min_{x,y,z} x^2 + 2y^2 + 4z^2.$$

Konstruišemo MATLAB funkciju koja implementira metod najbržeg pada sa linijskim pretraživanjem unazad. U nastavku dajemo njen kod.

```
function [x,fun_vr] = gm_unazad(f,g,x0,s,alfa,beta,epsilon)
%Gradijentni metod sa linijskim pretrazivanjem unazad
%
%INPUT
%============================
%f...........funkcija cilja
%g...........gradijent funkcije cilja
%x0..........inicijalna tacka
%s...........inicijalni izbor koraka
%alfa........parametar tolerancije za selekciju koraka
%beta........0<beta<1
%epsilon.....prag tolerancije
%OUTPUT
%============================
%x...........optimalno resenje min f(x) uz prag tolerancije
%fun_vr......optimalna vrednost funkcije cilja uz prag tolerancije


x=x0;
iter=0;
grad=g(x);
fun_vr=f(x);
while (norm(grad) > epsilon) && (iter < 100)
    iter=iter+1;
    t=s;
    while (fun_vr - f(x-t*grad) < alfa*t*norm(grad)^2)
        t=beta*t;
    end
    x=x-t*grad;
    fun_vr=f(x);
    grad=g(x);
    fprintf('broj_iter = %3d norma_grad = %2.6f fun_vr = %e \n',...
        iter,norm(grad),fun_vr);
end
```

Pre nego što pozovemo funkciju treba da definišemo ulazne varijable. Naime, neka početni vektor bude isti kao i do sad $x_0 = (2, 1, 1)^T$ i neka su parametri sledeći $\varepsilon = 10^{-5}, s = 2, \alpha = \frac{1}{4}$ i $\beta = \frac{1}{2}$. Rezultat nakon pozivanja funkcije je sledeći:

```
» A=[1,0,0;0,2,0;0,0,4];
» [x,fun_vr]=gm_unazad(@(x)x'*A*x,@(x)2*A*x,[2;1;1],2,0.25,0.5,
1.e-5)
```

broj_iter = 1 norma_grad = 3.605551 fun_vr = 2.750000e+00
broj_iter = 2 norma_grad = 2.000000 fun_vr = 5.000000e-01
broj_iter = 3 norma_grad = 0.000000 fun_vr = 0.000000e+00

```
x =
   0
   0
   0
fun_vr =
   0
```

□

Primetimo da se metod najbržeg pada sa linijskim pretraživanjem unazad završio nakon samo 3 iteracije. Pri tome konvergira ka tačnom optimalnom rešenju što u opštem slučaju ne mora da važi.

U navedenim primerima videli smo da nema prednosti u izvodjenju tačnog linijskog pretraživanja, šta više, bolje rezultate smo dobili sa netačnim linijskim pretraživanjem.

Da metod najbržeg pada zaista može da ima loše osobine pokazujemo u sledećem primeru. Posmatrajmo problem minimizacije

$$\min_{x,y} \left\{50x^2 + \tfrac{1}{500}y^2\right\}$$

i primenimo metod najbržeg pada sa netačnim linijskim pretraživanjem (unazad) sa početnim vektorom $(1,1)^T$, a ostali parametri neka ostanu nepromenjeni.
Naime, izvršavamo sledeću MATLAB komandu:

```
» A=[50,0;0,0.05];
» [x,fun_vr]=gm_unazad(@(x)x'*A*x,@(x)2*A*x,[1;1],2,0.25,0.5,
1.e-5)
```

Dobijeni rezultat je oblika:

broj_iter = 1 norma_grad = 21.875228 fun_vr = 2.442500e+00
broj_iter = 2 norma_grad = 4.786198 fun_vr = 1.643325e-01
broj_iter = 3 norma_grad = 1.051497 fun_vr = 5.524454e-02
: :
broj_iter = 99 norma_grad = 0.115530 fun_vr = 3.229819e-02
broj_iter = 100 norma_grad = 0.092812 fun_vr = 3.217388e-02

Jasno da je 100 iteracija velik broj koraka da bi se dobila dobra aproksimacija dvodimenzionalnog problema. Pri tome, naš algoritam se zaustavio jer je izlazni kriterijum koji se odnosi na maksimalan broj iteracija koji dozvoljavamo zadovoljen. Da bismo videli koliko je zaista neophodno iteracija za konvergenciju posmatranog problema, izvršavamo još jednom isti algoritam koristeći samo standardni izlazni kriterijum, $\|\nabla f(\boldsymbol{x_{k+1}})\| \leqslant \varepsilon$.

Dobijeni rezultat je oblika:

broj_iter = 1 norma_grad = 21.875228 fun_vr = 2.442500e+00
broj_iter = 2 norma_grad = 4.786198 fun_vr = 1.643325e-01
broj_iter = 3 norma_grad = 1.051497 fun_vr = 5.524454e-02
: :
broj_iter = 4099 norma_grad = 0.000012 fun_vr = 3.622142e-10
broj_iter = 4100 norma_grad = 0.000010 fun_vr = 3.608277e-10

Glavno pitanje koje se postavlja, jeste, da li možemo da nađemo veličinu koja u izvesnom smislu može da predvidi neophodan broj iteracija u primeni metod najbržeg pada da bi taj metod konvergirao. Samim tim bi taj broj govorio i o tome koliko je sam problem kompleksan. Ovo jeste jedan od bitnijih problema čiji delimičan odgovor može da se nađe u *uslovnim brojevima*.

3.2 Uslovni brojevi

Posmatramo kvadratni problem minimizacije

$$\min_{\boldsymbol{x}\in\mathbb{R}^{\boldsymbol{n}}} \left\{ f(\boldsymbol{x}) \equiv \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} \right\}, \tag{3.3}$$

gde je $\boldsymbol{A} > \boldsymbol{0}$. Trivijalno, optimalno rešenje problema je $\boldsymbol{x}^* = 0$. Forma metode najbržeg pada sa tačnim linijskim pretraživanjem je oblika $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - t_k \boldsymbol{d_k}$, pri tom $\boldsymbol{d_k} = 2\boldsymbol{A}\boldsymbol{x_k}$ je gradijent funkcije f u tački $\boldsymbol{x_k}$. Kao što smo već izvodili, u slučaju kvadratne funkcije i primene metode najbržeg pada sa tačnim linijskim pretraživanjem na istu, dolazimo do eksplicitnog izraza koraka t_k, tako u ovom primeru korak t_k je oblika $t_k = \dfrac{\boldsymbol{d_k}^T \boldsymbol{d_k}}{2\boldsymbol{d_k}^T \boldsymbol{A}\boldsymbol{d_k}}$. Dakle,

$$\begin{aligned} f(\boldsymbol{x_{k+1}}) &= \boldsymbol{x_{k+1}}^T \boldsymbol{A}\boldsymbol{x_{k+1}} \\ &= (\boldsymbol{x_k} - t_k\boldsymbol{d_k})^T \boldsymbol{A}(\boldsymbol{x_k} - t_k\boldsymbol{d_k}) \\ &= \boldsymbol{x_k}^T \boldsymbol{A}\boldsymbol{x_k} - 2t_k \boldsymbol{d_k}^T \boldsymbol{A}\boldsymbol{x_k} + {t_k}^2 \boldsymbol{d_k}^T \boldsymbol{A}\boldsymbol{d_k} \\ &= \boldsymbol{x_k}^T \boldsymbol{A}\boldsymbol{x_k} - t_k \boldsymbol{d_k}^T \boldsymbol{d_k} + {t_k}^2 \boldsymbol{d_k}^T \boldsymbol{A}\boldsymbol{d_k}. \end{aligned}$$

Ako u poslednjoj jednačini zamenimo t_k i ponovo iskoristimo da je $\boldsymbol{d_k} = 2\boldsymbol{A}\boldsymbol{x_k}$, imamo

$$\begin{aligned} f(\boldsymbol{x_{k+1}}) &= \boldsymbol{x_k}^T \boldsymbol{A}\boldsymbol{x_k} - \frac{1}{4}\frac{(\boldsymbol{d_k}^T \boldsymbol{d_k})^2}{\boldsymbol{d_k}^T \boldsymbol{A}\boldsymbol{d_k}} \\ &= \boldsymbol{x_k}^T \boldsymbol{A}\boldsymbol{x_k}\Big(1 - \frac{1}{4}\frac{(\boldsymbol{d_k}^T \boldsymbol{d_k})^2}{(\boldsymbol{d_k}^T \boldsymbol{A}\boldsymbol{d_k})(\boldsymbol{x_k}^T \boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{x_k})}\Big) \\ &= \boldsymbol{x_k}^T \boldsymbol{A}\boldsymbol{x_k}\Big(1 - \frac{(\boldsymbol{d_k}^T \boldsymbol{d_k})^2}{(\boldsymbol{d_k}^T \boldsymbol{A}\boldsymbol{d_k})(2\boldsymbol{x_k}^T \boldsymbol{A}\boldsymbol{A}^{-1}2\boldsymbol{A}\boldsymbol{x_k})}\Big) \\ &= \Big(1 - \frac{(\boldsymbol{d_k}^T \boldsymbol{d_k})^2}{(\boldsymbol{d_k}^T \boldsymbol{A}\boldsymbol{d_k})(\boldsymbol{d_k}^T \boldsymbol{A}^{-1}\boldsymbol{d_k})}\Big) f(\boldsymbol{x_k}) \end{aligned} \tag{3.4}$$

Primenom Kantorovičeve nejednakosti poslednji izraz (3.4) postaje

$$f(\boldsymbol{x_{k+1}}) \leqslant \Big(1 - \frac{4Mm}{(M+m)^2}\Big) f(\boldsymbol{x_k}) = \Big(\frac{M-m}{M+m}\Big)^2 f(\boldsymbol{x_k}),$$

gde je $M = \lambda_{max}(\boldsymbol{A})$ i $m = \lambda_{min}(\boldsymbol{A})$. Sumiranje ove diskusije navodimo u sledećoj lemi.

Lema 3.2.1. *Neka je $\{x_k\}_{k\geqslant 0}$ niz generisan opadajućim gradijentim metodom sa tačnim linijskim pretraživanjem za rešavanje problema* (3.3)*. Tada za svako $k = 0, 1, 2...$ važi*

$$f(\boldsymbol{x}_{k+1}) \leqslant \Big(\frac{M-m}{M+m}\Big)^2 f(\boldsymbol{x_k}), \tag{3.5}$$

gde je $M = \lambda_{max}(\boldsymbol{A})$ i $m = \lambda_{min}(\boldsymbol{A})$.

Nejednakost (3.5) implicira

$$f(\boldsymbol{x_k}) \leqslant c^k f(\boldsymbol{x_0}),$$

gde je $c = \big(\frac{M-m}{M+m}\big)^2$. Drugim rečima, niz vrednosti funkcija ograničeno je od gore sa opadajućim geometrijskim nizom. U ovom slučaju kažemo da niz vrednosti funkcija konvergira sa linearnim koeficijentom ka optimalnoj vrednosti. Brzina konvergencije zavisi od c, naime kako c postaje veće, brzina konvergencije se smanjuje. Veličina c drugačije može da se zapiše kao

$$c = \left(\frac{\chi - 1}{\chi + 1}\right)^2,$$

gde je $\chi = \dfrac{\lambda_{max}(\boldsymbol{A})}{\lambda_{min}(\boldsymbol{A})} = \dfrac{M}{m}$. Kako je c rastuća funkcija po χ, sledi da ponašanje metode najbržeg pada zavisi od količnika maksimalnog i minimalnog karakterističnog korena matrice $\boldsymbol{A}$, te taj broj nazivamo *uslovni broj*. Iako se uslovni broj može definisati za generalne matrice, mi se u ovom radu ograničavamo na pozitivno definitne matrice, te u nastavku dajemo definiciju uslovnog broja za takvu klasu matrica.

Definicija 3.2.1. Neka je $\boldsymbol{A}$ $n \times n$ pozitivno definitna matrica. Tada vrednost

$$\chi(\boldsymbol{A}) = \frac{\lambda_{max}(\boldsymbol{A})}{\lambda_{min}(\boldsymbol{A})}.$$

nazivamo ***uslovnim brojem*** matrice $\boldsymbol{A}$.

Naime, već smo ilustrovali u prethodnim primerima da metod najbržeg pada primenjem na probleme sa velikim uslovnim brojem može da zahteva veći broj iteracija, i obrnuto, metod najbržeg pada primenjen na probleme sa malim uslovnim brojem će verovatno konvergirati sa manjim brojem iteracija. Zaista, uslovni broj matrice problema $50x^2 + 0.05y^2$ je $\chi = 1000$, što je prilično velik broj, te je očekivan velik broj iteracija za konvergenciju, dok uslovni broj matrice problema $x^2 + 2y^2 + 4z^2$ jeste $\chi = 3$ što je znatno manji broj, te je očekivan i mali broj iteracija. Matrice sa malim uslovnim brojem nazivamo *dobro-uslovljene*, a matrice sa velikim uslovnim brojem nazivamo *loše-uslovljene* matrice. Imajući u vidu da je sva diskusija do sad vezana za restriktivnu klasu kvadratnih funkcija kada je Hesijan konstanta, pojam uslovnog broja se primenjuje u daleko široj klasi. U tom slučaju, poznato je da stopa konvergencije niza $\boldsymbol{x_k}$ za datu stacionarnu tačku $\boldsymbol{x}^*$ zavisi od uslovnog broja $\chi(\nabla^2 f(\boldsymbol{x}^*))$. Iako se u ovom radu nećemo baviti ovim teoretskim rezultatom, u nastavku ilustrujemo primer poznatog loše-uslovljenog problema.

Primer 3.2.1. *Data nam je Rozenbrok funkcija koja ima oblik:*

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Optimalno rešenje je očigledno $(x_1, x_2) = (1, 1)$ *sa odgovarajućom optimalnom vrednošću 0. Rozenbrok funkcija je ekstremno loše uslovljena u optimalnom rešenju. Zaista,*

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{pmatrix}$$
$$\nabla^2 f(\boldsymbol{x}) = \begin{pmatrix} -400x_2 + 1200x_1^2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}.$$

Nije teško pokažati da je $(x_1, x_2) = (1, 1)$ *jedinstvena stacionarna tačka. Ako Hesijan zamenimo sa stacionarnom tačkom, dobijamo:*

$$\nabla^2 f(1, 1) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}$$

te je uslovni broj:

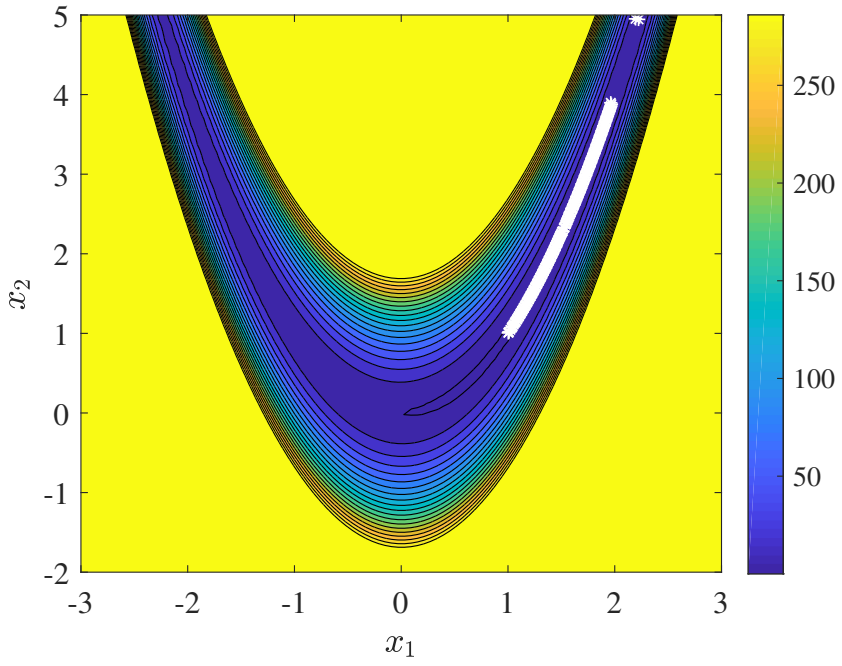
```
» H=[802,-400;-400,200];
» cond(H)
ans =
   2508.00
```

Uslovni broj veći od 2500 bi trebao da ima ozbiljne efekte na brzinu konvergencije gradijetnog metoda. Iz tog razloga primenjujemo metod najbržeg pada sa linijskim pretraživanjem unazad na Rozenbrok funkciju sa početnom tačkom $x_0 = (2, 5)^T$ i standardnim izlaznim kriterijumom $\|\nabla f(\boldsymbol{x_{k+1}})\| \leqslant \varepsilon$:

```
» [x,fun_vr]=gm_unazad(@(x)100*(x(2)-x(1)^2)^2+(1-x(1))^2,...
@(x)[-400*(x(2)-x(1)^2)*x(1)-2*(1-x(1));200*(x(2)-x(1)^2)],[2;5],
2,0.25,0.5,1.0000e-05);
```

broj_iter = 1 norma_grad = 118.254478 fun_vr = 3.221022
broj_iter = 2 norma_grad = 0.723051 fun_vr = 1.496586
: :
broj_iter = 6888 norma_grad = 0.000011 fun_vr = 9.596540e-11
broj_iter = 6889 norma_grad = 0.000020 fun_vr = 9.579907e-11
broj_iter = 6890 norma_grad = 0.000009 fun_vr = 9.557402e-11

Ovo izvršavanje zahtevalo je ogroman broj iteracije, te je efekat loše-uslovljenosti imao značajan uticaj. Da bismo bolje razumeli prirodu loše-uslovljenosti dajemo vizuelni prikaz kontura (linije dobijene rešavanjem jednačine $f(\boldsymbol{x}) = c$, pri tome je c neka konstanta) Rozenbrok funkcije sa iteracijama na Slici 3.1.



Slika 3.1: Konturne linije Rozenbrok funkcije sa hiljadama iteracija metode najbržeg pada.

□

3.3 Dijagonalno skaliranje

Loše uslovljeni problemi su jedan od najčešćih problema koji se javljaju u praksi, te su mnoge metode razvijene kako bi ga zaobišle. Jedan od najpoznatijih pristupa je da se uslovi problem tako što se napravi odgovarajuća linearna transformacija varijable na koju možemo da utičemo, odnosno $\boldsymbol{x}$. Preciznije, posmatrajmo neograničeni problem minimizacije

$$\min\{f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^n\}.$$

Za datu nesingularnu matricu $\boldsymbol{S} \in \mathbb{R}^{n\times n}$, pravimo linearnu transformaciju $\boldsymbol{x} = \boldsymbol{Sy}$, te dobijamo problem ekvivalentan početnom

$$\min\{g(\boldsymbol{y}) \equiv f(S\boldsymbol{y}) : \boldsymbol{y} \in \mathbf{R^n}\}.$$

Kako je $\nabla g(\boldsymbol{y}) = \boldsymbol{S}^T\nabla f(S\boldsymbol{y}) = \boldsymbol{S}^T\nabla f(\boldsymbol{x})$, sledi da metod najbržeg pada primenjen na transformisan problem ima oblik

$$\boldsymbol{y_{k+1}} = \boldsymbol{y_k} - t_k\boldsymbol{S}^T\nabla f(\boldsymbol{Sy_k}).$$

Množeći poslednju jednakost sa $\boldsymbol{S}$ s leve strane dobijamo

$$\boldsymbol{Sy_{k+1}} = \boldsymbol{Sy_k} - t_k\boldsymbol{SS}^T\nabla f(\boldsymbol{Sy_k}).$$

Koristeći notaciju $\boldsymbol{x_k} = \boldsymbol{Sy_k}$, dobijamo rekurzivnu formulu

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - t_k\boldsymbol{SS}^T\nabla f(\boldsymbol{x_k}).$$

Neka je $\boldsymbol{B} = \boldsymbol{SS}^T$. Definisaćemo sledeću verziju metode najbržeg pada, koju nazivamo *skalirani gradijentni metod* sa skaliranom matricom $\boldsymbol{B}$:

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - t_k\boldsymbol{B}\nabla f(\boldsymbol{x_k}).$$

Po svojoj definisanosti, matrica $\boldsymbol{B}$ je pozitivno definitna. Pravac $-\boldsymbol{B}\nabla f(\boldsymbol{x_k})$ je opadajući pravac funkcije f u $\boldsymbol{x_k}$ kada je $\nabla f(\boldsymbol{x_k}) \neq \mathbf{0}$ jer

$$f'(\boldsymbol{x_k}; -\boldsymbol{B}\nabla f(\boldsymbol{x_k})) = -\nabla f(\boldsymbol{x_k})^T\boldsymbol{B}\nabla f(\boldsymbol{x_k}) < 0,$$

gde poslednja stroga nejednakost sledi iz pozitivne definitnosti matrice $\boldsymbol{B}$. Sumiranjem prethodne diskusije, pokazali smo da skalirani gradijentni metod sa skaliranom matricom $\boldsymbol{B}$ je ekvivalentan metodu najbržeg pada primenjenom na funkciju $g(\boldsymbol{y}) = f(\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{y})$. Primetimo da su gradijent i Hesijan funkcije g dati sa

$$\nabla g(\boldsymbol{y}) = \boldsymbol{B}^{\frac{1}{2}}\nabla f(\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{y}) = \boldsymbol{B}^{\frac{1}{2}}\nabla f(\boldsymbol{x}),$$

$$\nabla^2 g(\boldsymbol{y}) = \boldsymbol{B}^{\frac{1}{2}}\nabla^2 f(\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{y})\boldsymbol{B}^{\frac{1}{2}} = \boldsymbol{B}^{\frac{1}{2}}\nabla^2 f(\boldsymbol{x})\boldsymbol{B}^{\frac{1}{2}},$$

gde je $\boldsymbol{x} = \boldsymbol{B}^{\frac{1}{2}}\boldsymbol{y}$.

U nastavku dajemo šematski prikaz skaliranog gradijentnog metoda.

Algoritam* : *Skalirani gradijentni metod

***Korak* 0.** Definisati ulazni parametar tolerancije, ε.
***Korak* 1.** Izabrati početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$.
***Korak* 2.** Izabrati skaliranu matricu $\boldsymbol{B_k} \succ \mathbf{0}$.
***Korak* 3.** Izabrati korak t_k metodom linijskog pretraživanja funkcije

$$g(t) = f(\boldsymbol{x_k} - t\boldsymbol{B_k}\nabla f(\boldsymbol{x_k})).$$

***Korak* 4.** Odrediti narednu iteraciju $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - t_k\boldsymbol{B_k}\nabla f(\boldsymbol{x_k})$.
***Korak* 5.** Ako je $\|\nabla f(\boldsymbol{x_{k+1}})\| < \varepsilon$, tada zaustavljamo iterativni postupak, izlazni parametar je $\boldsymbol{x_{k+1}}$. U suprotnom, uzeti da je $k = k + 1$ i ići na Korak 2.

Prirodno se postavlja pitanje, kako izabrati skaliranu matricu $\boldsymbol{B}_k$? Da bi ubrzali stopu konvergencije generisanog niza, koji zavisi od uslovnog broja skaliranog Hesijana $\boldsymbol{B}^{\frac{1}{2}}\nabla^2 f(\boldsymbol{x_k})\boldsymbol{B}^{\frac{1}{2}}$, skaliranu matricu najčešće biramo tako da skaliranog Hesijana učini da bude što bliži jediničnoj matrici. Kada je $\nabla^2 f(\boldsymbol{x_k}) \succ \boldsymbol{0}$, tada zapravo biramo da je $\boldsymbol{B}_k = (\nabla^2 f(\boldsymbol{x_k}))^{-1}$, te skalirani Hesijan postaje jedinična matrica. U tom slučaju dobijamo Njutnov metod

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - t_k(\nabla^2 f(\boldsymbol{x_k}))^{-1}\nabla f(\boldsymbol{x_k}).$$

Mane Njutnovog metoda jesu što nam je neophodno potpuno znanje Hesijana, te izraz $(\nabla^2 f(\boldsymbol{x_k}))^{-1}\nabla f(\boldsymbol{x_k})$ sugeriše na to da linearni sistem oblika $\nabla^2 f(\boldsymbol{x_k})\boldsymbol{d} = \nabla f(\boldsymbol{x_k})$ treba da bude izračunat u svakoj iteraciji, što može biti jako 'skupo' sa računarske tačke gledišta. Iz tih razloga predlažu se jednostavnije skalirane matrice. Najednostavniji izbor skaliranih matrica su upravo dijagonalne matrice. Prirodni izbor za dijagonalni elemenat matrice jeste

$$\boldsymbol{B}_{ii} = \left(\nabla^2 f(\boldsymbol{x_k})^{-1}\right)_{ii}.$$

Sa gore navedenim izborom za dijagonalne elemente, dijagonalni elementi matrice $\boldsymbol{B}^{\frac{1}{2}}\nabla^2 f(\boldsymbol{x_k})\boldsymbol{B}^{\frac{1}{2}}$ su sve jedinice. Naravno, takav izbor možemo napraviti samo kad je dijagonala Hesijana pozitivna.

Primer 3.3.1. *Posmatramo problem*

$$\min\{5000x_1^2 + 40x_1x_2 + x_2^2\}.$$

Počinjemo primenom metode najbržeg pada sa tačnim linijskim pretraživanjem, gde nam je inicijalna tačka $(1, 1000)^T$ *i parametar tolerancije* $\varepsilon = 10^{-5}$ *tako što primenjujemo MATLAB funkciju* `gmk`.

```
» A=[5000,20;20,1];
» gmk(A,[0;0],[1;1000],1e-5)
```

```
broj_iter = 1 norma_grad = 1840.854230 fun_vr = 9.196245e+05
broj_iter = 2 norma_grad = 44037.778062 fun_vr = 8.092910e+05
broj_iter = 3 norma_grad = 1619.994807 fun_vr = 7.121951e+05
broj_iter = 4 norma_grad = 13912.003797 fun_vr = 113948.378627
                    :              :
broj_iter = 297 norma_grad = 0.000011 fun_vr = 3.419010e-11
broj_iter = 298 norma_grad = 0.000269 fun_vr = 3.008809e-11
broj_iter = 299 norma_grad = 0.000010 fun_vr = 2.647822e-11

ans =
    1.0e-05 *

    -0.002149931507966
    0.536476182784195
```

Velik broj iteracija nije iznenadjujući jer je uslovni broj matrice problema velik:

```
» cond(A)
ans =

    5434.95
```

Skalirani gradijentni metod sa dijagonalnom skaliranom matricom

$$\boldsymbol{B}=\begin{pmatrix} \frac{1}{5000} & 0 \\ 0 & 1 \end{pmatrix}$$

trebao bi brže da konvergira jer je uslovni broj skalirane matrice $\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{\frac{1}{2}}$ značajno manji:

```
» B=diag(1./diag(A));
» S=sqrtm(B)*A*sqrtm(B)
S =
   1.000000000000000   0.282842712474619
   0.282842712474619   1.000000000000000
» cond(S)
ans =
   1.788788505379606
```

Da bismo proverili performanse skaliranog gradijentnog metoda, koristićemo modifikovanu MATLAB funkciju `gmk`, koju nazivamo `gradijent_skalirana_kvadratna`.

```
function [x,fun_vr]=gradijent_skalirana_kvadratna(A,b,B,x0,epsilon)
% INPUT
% ======================
% A ....... pozitivno definitivna matrica funkcije cilja
% b ....... vektor kolona linearnog oblika problema minimizacije
% B ....... skalirana matrica
% x0 ...... pocetna tacka metoda
% epsilon . parametar tolerancije
% OUTPUT
% ======================
% x ....... optimalno resenje uz prag tolerancije
% fun_val . optimalna vrednost funkcije uz prag tolerancije

x=x0;
iter=0;
grad=2*(A*x+b);
while (norm(grad) > epsilon) && (iter < 100)
iter=iter+1;
t=grad'*B*grad/(2*(grad'*B')*A*(B*grad));
x=x-t*B*grad;
grad=2*(A*x+b);
fun_vr=x'*A*x+2*b'*x;
fprintf('iter_number = %3d norm_grad = %2.6f fun_vr = %e \n',...
iter,norm(grad),fun_vr);
end
```

Izvršavanjem ovog koda zahteva se samo 16 iteracija:

```
» gradijent_skalirana_kvadratna(A,[0;0],B,[1;1000],1e-5)
```

broj_iter = 1 norma_grad = 27273.819719 fun_vr = 5.321005e+04
broj_iter = 2 norma_grad = 2548.053472 fun_vr = 2.709387e+03
broj_iter = 3 norma_grad = 1388.747619 fun_vr = 1.379585e+02
: :
broj_iter = 14 norma_grad = 0.000044 fun_vr = 8.229991e-13
broj_iter = 15 norma_grad = 0.000024 fun_vr = 4.190605e-14
broj_iter = 16 norma_grad = 0.000002 fun_vr = 2.133802e-15

```
ans =
   1.0e-07 *
    0.000451875603652
    0.451875603651910
```

□

3.4 Gaus-Njutnov metod

Posmatramo nelinearni problem najmanjih kvadrata

$$\min_{\boldsymbol{x}\in\mathbb{R}^n}\Big\{g(\boldsymbol{x})\equiv\sum_{n=1}^{m}f_i(\boldsymbol{x}-c_i)^2\Big\}. \tag{3.6}$$

Pretpostavićemo da su $f_1,...,f_m$ neprekidno diferencijabilne na celom $\mathbb{R}^n$ za svako $i=1,2,...m$ i da su $c_1,...,c_m\in\mathbb{R}^n$. Problem se može zapisati i u vektorskom obliku. Definišemo funkciju

$$\boldsymbol{F}(\boldsymbol{x})=\begin{pmatrix}f_1(\boldsymbol{x})-c_1\\ f_2(\boldsymbol{x})-c_2\\ \vdots\\ f_m(\boldsymbol{x})-c_m\end{pmatrix},$$

tada posmatrani problem dobija oblik

$$\min\|\boldsymbol{F}(\boldsymbol{x})\|^2.$$

Generalni korak Gaus-Njutnovog metoda glasi: za dato k izračunaj $\boldsymbol{x_k}$, a sledeću iteraciju izaberi tako da minimizira sumu kvadrata linearne aproksimacije f_i u tački $\boldsymbol{x_k}$ što je

$$\boldsymbol{x_{k+1}}=\underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}}\Big\{\sum_{i=1}^{m}\big[f_i(\boldsymbol{x_k})+\nabla f_i(\boldsymbol{x_k})^T(\boldsymbol{x}-\boldsymbol{x_k})-c_i\big]^2\Big\}. \tag{3.7}$$

Navedeni problem minimizacije je u suštini linearni problem najmanjih kvadrata

$$\min_{\boldsymbol{x}\in\mathbb{R}^n}\|\boldsymbol{A}_k\boldsymbol{x}-\boldsymbol{b_k}\|,$$

gde je

$$\boldsymbol{A}_k=\begin{pmatrix}\nabla f_1(\boldsymbol{x_k})^T\\ \nabla f_2(\boldsymbol{x_k})^T\\ \vdots\\ \nabla f_m(\boldsymbol{x_k})^T\end{pmatrix}=\boldsymbol{J}(\boldsymbol{x_k})$$

matrica Jakobijana i

$$\boldsymbol{b_k}=\begin{pmatrix}\nabla f_1(\boldsymbol{x_k})^T\boldsymbol{x_k}-f_1(\boldsymbol{x_k})+c_1\\ \nabla f_2(\boldsymbol{x_k})^T\boldsymbol{x_k}-f_2(\boldsymbol{x_k})+c_2\\ \vdots\\ \nabla f_m(\boldsymbol{x_k})^T\boldsymbol{x_k}-f_m(\boldsymbol{x_k})+c_m\end{pmatrix}=\boldsymbol{J}(\boldsymbol{x_k})\boldsymbol{x_k}-\boldsymbol{F}(\boldsymbol{x_k}).$$

Pretpostavljamo da je matrica Jakobijana punog ranga, jer drugačije minimizacija problema (3.6) ne bi imala jedinstveni minimizator (Napomena: poznato nam je iz literature [2] da ako je matrica problema najmanjih kvadrata punog ranga tada je Hesijan pozitivno definitna matrica, što po Lemi 2.6.1. sledi da postoji jedinstveni minimizator). U tom slučaju možemo i eksplicitno da zapišemo Gaus-Njutnove iteracije:

$$\boldsymbol{x_{k+1}} = (\boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{J}(\boldsymbol{x_k}))^{-1} \boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{b_k}.$$

Metod može biti zapisan i kao

$$\begin{aligned} \boldsymbol{x_k} &= (\boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{J}(\boldsymbol{x_k}))^{-1} \boldsymbol{J}(\boldsymbol{x_k})^T (\boldsymbol{J}(\boldsymbol{x_k})\boldsymbol{x_k} - \boldsymbol{F}(\boldsymbol{x_k})) \\ &= \boldsymbol{x_k} - (\boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{J}(\boldsymbol{x_k}))^{-1} \boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{F}(\boldsymbol{x_k}). \end{aligned}$$

Stoga, Gaus-Njutnov pravac je $\boldsymbol{d_k} = (\boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{J}(\boldsymbol{x_k}))^{-1} \boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{F}(\boldsymbol{x_k})$. Uočimo da je $\nabla g(\boldsymbol{x}) = 2\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{F}(\boldsymbol{x})$, te možemo da zaključimo

$$\boldsymbol{d_k} = \frac{1}{2}(\boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{J}(\boldsymbol{x_k}))^{-1} \nabla g(\boldsymbol{x_k}) \ ,$$

što znači da je Gaus-Njutnov metod, u osnovi, skalirani gradijentni metod sa sledećom pozitivno definitnom skaliranom matricom

$$\boldsymbol{B} = \frac{1}{2}(\boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{J}(\boldsymbol{x_k}))^{-1}.$$

Ova činjenica takođe objašnjava zašto je Gaus-Njutnov metod ustvari metod opadajućih pravca. Metod koji smo sada opisali poznat je pod nazivom *čist Gaus-Njutnov metod* i kod njega nije uključen korak t_k linijskog pretraživanja funkcije $g(\boldsymbol{x_k} - t\boldsymbol{d_k})$. Modifikacijom ovog metoda, tako što uključujemo i korak, dobijamo *prigušeni Gaus-Njutnov metod*.

Algoritam : Prigušeni Gaus − Njutnov metod

***Korak* 0.** Definisati ulazni parametar tolerancije, ε.
***Korak* 1.** Izabrati početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$.
***Korak* 2.** Postaviti da je $\boldsymbol{d_k} = (\boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{J}(\boldsymbol{x_k}))^{-1} \boldsymbol{J}(\boldsymbol{x_k})^T \boldsymbol{F}(\boldsymbol{x_k})$.
***Korak* 3.** Izabrati korak t_k metodom linijskog pretraživanja, to jest:

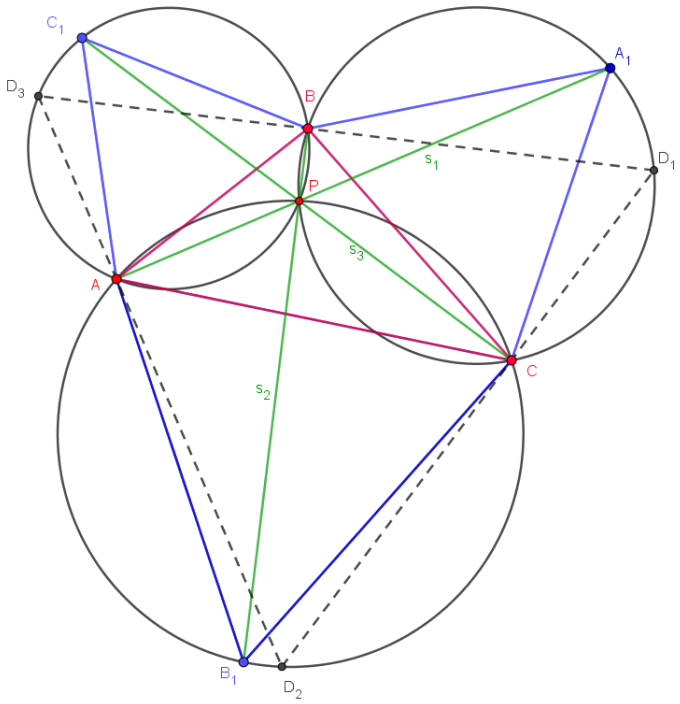
$$t_k \in \underset{t \geqslant 0}{\operatorname{argmin}}\, g(\boldsymbol{x_k} - t\boldsymbol{d_k}).$$

***Korak* 4.** Odrediti narednu iteraciju $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - t_k \boldsymbol{d_k}$.
***Korak* 5.** Ako je $\|\nabla g(\boldsymbol{x_{k+1}})\| \leqslant \varepsilon$, tada zaustavljamo iterativni postupak, izlazni parametar je $\boldsymbol{x_{k+1}}$. U suprotnom, uzeti da je $k = k + 1$ i ići na Korak 2.

3.5 Ferma-Veberov problem

Gradijentni metod je baza za mnoge druge metode koje na prvi pogled ne deluju da imaju veze sa navedenim metodom. Jedan od takvih interesantnih metoda nalazi se u teoriji lokacije kao što je Ferma-Veberov problem. Francuski matematičar Pjer de Ferma (1601-1665) se najčešće navodi kao prvi koji je postavio jedan lokacijski problem: naći tačku trougla čiji je zbir rastojanja do temena minimalan. Tu tačku je nazvao petom značajnom tačkom trougla. Galilejev učenik, Toričeli (1608-1647) navodi se kao prvi koji je uspeo da reši problem konstruktivno: konstruisati jednakostranične trouglove nad svakom stranicom, sa trećim temenom sa suprotne strane od unutrašnjosti početnog trougla (Slika 3.2). Tražena tačka (Toričelijeva tačka) nalazi se na preseku krugova opisanih oko jednakostraničnih trouglova.

Batista Kavalieri (1598-1647) pokazuje da duži koje spajaju Toričelijevu tačku s temenima trougla, obrazuju ugao od 120 stepeni. Sto godina kasnije (1750), engleski



Slika 3.2: Konstruktivna rešenja nalaženja pete značajne tačke trougla (A,B i C - fiksne tačke, P - Toričelijeva tačka).

matematičar Tomas Simpson (1710-1761) predlaže jednostavnije konstruktivno rešenje. On takodje koristi jednakostranične trouglove kao i Torčeli, ali traženu tačku nalazi kao presek pravih dobijenih spajanjem novih temena jednakostraničnih trouglova i datih temena sa suprotne strane (Slika 3.2). Ove prave su kasnije nazvane Simpsonove prave.

Takođe na istoj slici, konstruisan je i isprekidani trougao $D_1D_2D_3$ opisan oko trougla određenog sa tri proizvoljne tačke. On je zapravo rešenje sledećeg problema: naći najveći jednakostranični trougao opisan oko proizvoljnog trougla (tj. stranice traženog opisanog jednakostraničnog trougla prolaze kroz temena datog proizvoljnog trougla). Pri tome može se uočiti da su stranice traženog trougla normalne na Simpsonove prave. Kako je kasnije primetio Kun (1967), ovaj zadatak je dual problema koji je postavio Ferma: ako je poznata Torčijeva tačka (primal), tada je lako konstruisati najveći opisan jednakostranični trougao (dual), i obrnuto. Inače, ideja o dualnosti je fundamentalna u metodama optimizacije: svaki problem optimizacije ima svog parnjaka (dual) baziranog na suprotnostima (ili je neka vrsta dopune do celine).

U 20. veku, austrijski ekonomista Veber (1909) predlaže težinski model s tri tačke u problemu nalaženja lokacije fabrike, a s ciljem minimizacije transportnih troškova od fabrike do tri grupe snabdevača sirovinama. Interesantno je da je uopštenje ovog problema na m fiksnih tačaka vezano za Veberovo ime, iako ga on nije prvi ni predložio ni rešio, već je našao primenu u industriji. U ovom radu govorimo samo o Veberov problemu odnosno lokacijskom problemu jednog objekta, kome pristupamo sa matematičkog apekta i dajemo vezu sa gradijentnim metodom.

Za datih m tačaka iz $\mathbb{R}^n$: $\boldsymbol{a_1}, ..., \boldsymbol{a_m}$ (koje još nazivamo *usidrene* tačke) i m pondera $w_1, ..., w_m > 0$, naći tačku $\boldsymbol{x} \in \mathbb{R}^n$ za koju je suma težinskih rastojanja do datih tačaka $\boldsymbol{a_1}, ..., \boldsymbol{a_m}$ minimalna. Ako to formalno zapišemo, imamo problem minimizacije

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ f(\boldsymbol{x}) \equiv \sum_{i=1}^{m} w_i \|\boldsymbol{x} - \boldsymbol{a_i}\| \right\}.$$

Primetimo da funkcija cilja nije diferencijabilna u tačkama $\boldsymbol{a_1}, ..., \boldsymbol{a_m}$. Taj problem je jedan od fundamentalnih problema kada je reč o lokalizaciji, kao što je to slučaj u teoriji lokacije. Na primer, neka $\boldsymbol{a_1}, ...\boldsymbol{a_m}$ predstavljaju lokacije gradova, a $\boldsymbol{x}$ će biti lokacija aerodroma ili bolnice (ili bilo kog drugog objekta koji služi gradovima); dok ponderi mogu biti proporcionalni veličini stanovništva svakog grada. Jedan poznati pristup rešavanja ovakvog problema jeste Vajsfeldova metoda iz 1937. godine. Polazimo od uslova optimalnosti prvog reda

$$\nabla f(\boldsymbol{x}) = 0.$$

Pretpostavimo da $\boldsymbol{x}$ nije usidrena tačka. Navedena jednakost ekvivalentna je zapisu

$$\sum_{i=1}^{m} w_i \frac{1}{2} \frac{2(\boldsymbol{x} - \boldsymbol{a_i})}{\sqrt{(\boldsymbol{x} - \boldsymbol{a_i})^T (\boldsymbol{x} - \boldsymbol{a_i})}} = 0,$$

odakle sledi

$$\sum_{i=1}^{m} w_i \frac{\boldsymbol{x} - \boldsymbol{a_i}}{\|\boldsymbol{x} - \boldsymbol{a_i}\|} = 0.$$

Potom lako dobijamo da je

$$\Big(\sum_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x} - \boldsymbol{a_i}\|} \Big) \boldsymbol{x} = \sum_{i=1}^{m} \frac{w_i \boldsymbol{a_i}}{\|\boldsymbol{x} - \boldsymbol{a_i}\|},$$

odakle je

$$\boldsymbol{x} = \frac{1}{\sum\limits_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x} - \boldsymbol{a_i}\|}} \sum_{i=1}^{m} \frac{w_i \boldsymbol{a_i}}{\|\boldsymbol{x} - \boldsymbol{a_i}\|}.$$

Možemo preformulisati uslov optimalnosti kao $\boldsymbol{x} = T(\boldsymbol{x})$, gde je T operator

$$T(\boldsymbol{x}) \equiv \frac{1}{\sum\limits_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x} - \boldsymbol{a_i}\|}} \sum_{i=1}^{m} \frac{w_i \boldsymbol{a_i}}{\|\boldsymbol{x} - \boldsymbol{a_i}\|}.$$

Prema tome, problem traženja stacionarne tačke funkcije f može biti preoblikovan kao problem traženja fiksne tačke operatora T. Dakle, prirodan pristup rešavanja problema je preko metoda fiksne tačke, što je po iteracijama

$$\boldsymbol{x_{k+1}} = T(\boldsymbol{x_k}).$$

Sada dajemo eksplicitan zapis Vajsfeldove teoreme za rešavanje Ferma-Veberovog problema.

Algoritam : Vajsfeldov metod

***Korak* 0.** Izabrati početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$, tako da $\boldsymbol{x_0} \neq \boldsymbol{a_1}, ..., \boldsymbol{a_m}$.
***Korak* 1.** Za svako naredno k, odrediti iteraciju na sledeći način:

$$\boldsymbol{x_{k+1}} = T(\boldsymbol{x_k}) = \frac{1}{\sum\limits_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|}} \sum_{i=1}^{m} \frac{w_i \boldsymbol{a_i}}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|}.$$

Primetimo da je algoritam definisan samo kad su sve iteracije $\boldsymbol{x_k}$ različite od $\boldsymbol{a_1}, ..., \boldsymbol{a_m}$. Iako je algoritam inicijalno predstavljen kao metod fiksne tačke, on je u osnovi gradijentni metod. Zaista,

$$\begin{aligned}\boldsymbol{x_{k+1}} &= \frac{1}{\sum\limits_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|}} \sum_{i=1}^{m} \frac{w_i \boldsymbol{a_i}}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|} \\ &= \boldsymbol{x_k} - \frac{1}{\sum\limits_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|}} \sum_{i=1}^{m} w_i \frac{\boldsymbol{x_i} - \boldsymbol{a_i}}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|} \\ &= \boldsymbol{x_k} - \frac{1}{\sum\limits_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|}} \nabla f(\boldsymbol{x_k}).\end{aligned}$$

Dakle, Vajsfeldov metod je u suštini gradijentni metod sa specijalnim izborom koraka

$$t_k = \frac{1}{\sum\limits_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|}}.$$

Primer 3.5.1. *Daćemo, sada, jedan primer Vajsfeldovog metoda (sa jednakim ponderima) implementiranog u softverskom paketu MATLAB pod nazivom* `Vajsfeld`*. U nastavku sledi njegov kod.*

```
function y = Vajsfeld(A,x0,tol)
% INPUT
% ======================
% A ....... matrica dimenzije m x n koja sadrzi usidrene tacke
% x0 ...... pocetna tacka metoda
% tol ....... prag tolerancije
% OUTPUT
% ======================
% y ....... optimalno resenje

tol = 0.001 ; % prag tolerancije
[tacke, dim] = size(A) ;
eps = 1 ;
brojac = 0 ;

xt(1,:) = x0 ; %pocetna tacka

while eps > tol
    brojac = brojac + 1 ;
    w = sum((A - xt(brojac,:)).^2, 2).^-0.5 ;
    t = sum(A .* (w .* ones(tacke, dim)), 1) / sum(w) ;
    xt(brojac+1, :) = t ;
    eps = (sum((xt(brojac+1,:) - xt(brojac,:)).^2))^0.5 ;
end
y = xt(brojac+1,:);
```

Prvo posmatramo prostor $\mathbb{R}$. Uzimamo na primer četiri proizvoljne tačke i zanima nas ona tačka čije je rastojanje od datih tačaka minimalno. Drugim rečima, pozivamo sledeću MATLAB komandu:

```
» [y] = Vajsfeld([3;0;-4;7],0.5,0.001)
```

dobijeni izlaz koda je:

```
» y = 0.5
```

Posmatramo sada prostor $\mathbb{R}^3$. Ako uzmemo tri proizvoljne tačke tog prostora i izvršimo sledeću MATLAB komandu:

```
» [y]=Vajsfeld([2 -1 0; -1 2 -1;0 -1 2],[0 0 0],0.001)
```

dobijeni rezultat je:

```
» y = 0.6037 -0.4056 0.6037
```

Naravno, potrebno je i odgovoriti na nekoliko važnih pitanja. Kao prvo, treba utvrditi da li je ovaj metod dobro definisan? To bi značilo da možemo da garantuje da nijedna od iteracija $\boldsymbol{x_k}$ nije jednaka bilo kojoj tački od $\boldsymbol{a_1}, ..., \boldsymbol{a_m}$. Zatim, da li niz vrednosti funkcije cilja opada? Napokon, i da li niz $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ konvergira ka globalnom optimalnom rešenju? U ovom odeljku daćemo odgovore na neka od ovih pitanja.

Za početak, želimo da pokažemo da je generisani niz vrednosti funkcije nerastući, te definišemo pomoćnu funkciju

$$h(\boldsymbol{y}, \boldsymbol{x}) \equiv \sum_{i=1}^{m} w_i \frac{\|\boldsymbol{y} - \boldsymbol{a_i}\|^2}{\|\boldsymbol{x} - \boldsymbol{a_i}\|}, \boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{x} \in \mathbb{R}^n \backslash \mathscr{A},$$

gde je $\mathscr{A} \equiv \{\boldsymbol{a_1}, \boldsymbol{a_2}, ..., \boldsymbol{a_m}\}$. Funkcija $h(\cdot, \cdot)$ ima nekoliko bitnih osobina. Najpre, operator $\boldsymbol{T}$ može biti izračunat na vektoru $\boldsymbol{x} \notin \mathscr{A}$ minimiziranjem funkcije $h(\boldsymbol{y}, \boldsymbol{x})$ za svako $\boldsymbol{y} \in \mathbb{R}^n$.

Lema 3.5.1. *Za svako $\boldsymbol{x} \in \mathbb{R}^n \backslash \mathscr{A}$, važi*

$$T(\boldsymbol{x}) = \underset{y}{\operatorname{argmin}}\{h(\boldsymbol{y}, \boldsymbol{x}) : \boldsymbol{y} \in \mathbb{R}^n\}. \tag{3.8}$$

Dokaz. *Funkcija $h(\boldsymbol{y}, \boldsymbol{x})$ je kvadratna funkcija po promenljivoj $\boldsymbol{y}$, samim tim je njena matrica pozitivno definitna. U stvari, njena matrica je oblika $(\sum_{i=1}^{m} \frac{w_i}{\|\boldsymbol{x_k} - \boldsymbol{a_i}\|})\boldsymbol{I}$. Prema tome, po Lemi 2.6.1., jedinstveni globalni miminum problema (3.8), koji obeležavamo kao $\boldsymbol{y}^*$, ujedno je i jedinstvena stacionarna tačka funkcije $h(\cdot, \boldsymbol{x})$, što je tačka u kojoj gradijent nestaje:*

$$\nabla_y h(\boldsymbol{y}^*, \boldsymbol{x}) = \boldsymbol{0}.$$

Iz toga sledi

$$2 \sum_{i=1}^{m} w_i \frac{\boldsymbol{y}^* - \boldsymbol{a_i}}{\|\boldsymbol{x} - \boldsymbol{a_i}\|} = 0.$$

Izražavanjem $\boldsymbol{y}^*$ iz poslednje jednačine dobijamo da je $\boldsymbol{y}^* = T(\boldsymbol{x})$, i samim tim dolazimo do rešenja.

Poslednja lema nam u suštini kaze da Vajsfeldov metod može da se zapiše kao

$$\boldsymbol{x_{k+1}} = \operatorname{argmin}\{h(\boldsymbol{x}, \boldsymbol{x_k}) : \boldsymbol{x} \in \mathbb{R}^n\}.$$

Sada želimo da pokažemo još neke bitne osobine funkcije h koje će biti krucijalne za pokazivanje da je niz $\{f(\boldsymbol{x_k})\}_{k\geqslant 0}$ nerastući.

Lema 3.5.2. *Ako $\boldsymbol{x} \in \mathbb{R}^n \backslash \mathscr{A}$, onda*

(a) $h(\boldsymbol{x}, \boldsymbol{x}) = f(\boldsymbol{x})$,

(b) $h(\boldsymbol{y}, \boldsymbol{x}) \geqslant 2f(\boldsymbol{y}) - f(\boldsymbol{x})$ *za svako* $\boldsymbol{y} \in \mathbb{R}^n$,

(c) $f(T(\boldsymbol{x})) \leqslant f(\boldsymbol{x})$ *i* $f(T(\boldsymbol{x})) = f(\boldsymbol{x})$ *ako i samo ako* $\boldsymbol{x} = T(\boldsymbol{x})$,

(d) $\boldsymbol{x} = T(\boldsymbol{x})$ *ako i samo ako* $\nabla f(\boldsymbol{x}) = 0$.

Dokaz. *(a)* $h(\boldsymbol{x}, \boldsymbol{x}) = \sum_{i=1}^{m} w_i \dfrac{\|\boldsymbol{x} - \boldsymbol{a_i}\|^2}{\|\boldsymbol{x} - \boldsymbol{a_i}\|} = \sum_{i=1}^{m} w_i \|\boldsymbol{x} - \boldsymbol{a_i}\| = f(\boldsymbol{x})$.

(b) Za bilo koji nenegativan broj a i pozitivan broj b važi

$$(a-b)^2 \geqslant 0 \iff \frac{a^2}{b} \geqslant 2a - b.$$

Zamenjujući $a = \|\boldsymbol{y} - \boldsymbol{a_i}\|$ *i* $b = \|\boldsymbol{x} - \boldsymbol{a_i}\|$, *sledi da za svako* $i = 1, 2, ..., m$

$$\frac{\|\boldsymbol{y} - \boldsymbol{a_i}\|^2}{\|\boldsymbol{x} - \boldsymbol{a_i}\|} \geqslant 2\|\boldsymbol{y} - \boldsymbol{a_i}\| - \|\boldsymbol{x} - \boldsymbol{a_i}\|.$$

Množeći gornju jednačinu sa w_i *i sumiranjem iste po* $i = 1, 2, ..., m$, *dobijamo*

$$\sum_{i=1}^{m} w_i \frac{\|\boldsymbol{y} - \boldsymbol{a_i}\|^2}{\|\boldsymbol{x} - \boldsymbol{a_i}\|} \geqslant 2 \sum_{i=1}^{m} w_i \|\boldsymbol{y} - \boldsymbol{a_i}\| - \sum_{i=1}^{m} w_i \|\boldsymbol{x} - \boldsymbol{a_i}\|.$$

Dakle,

$$h(\boldsymbol{y}, \boldsymbol{x}) \geqslant 2f(\boldsymbol{y}) - f(\boldsymbol{x}).$$

(c) Kako je $T(\boldsymbol{x}) = \underset{y \in \mathbb{R}^n}{\operatorname{argmin}} h(\boldsymbol{y}, \boldsymbol{x})$, *sledi da je*

$$h(T(\boldsymbol{x}), \boldsymbol{x}) \leqslant h(\boldsymbol{x}, \boldsymbol{x}) = f(\boldsymbol{x}),$$

gde poslednja jednakost sledi iz (a). Iz dela (b) posmatrane leme imamo

$$h(T(\boldsymbol{x}), \boldsymbol{x}) \geqslant 2f(T(\boldsymbol{x})) - f(\boldsymbol{x}),$$

te poslednje dve nejednakosti impliciraju

$$f(\boldsymbol{x}) \geqslant h(T(\boldsymbol{x}), \boldsymbol{x}) \geqslant 2f(T(\boldsymbol{x})) - f(\boldsymbol{x}), \tag{3.9}$$

odakle zaključujemo da je $f(T(\boldsymbol{x})) \leqslant f(\boldsymbol{x})$. *Treba još da pokažemo da je* $f(T(\boldsymbol{x})) = f(\boldsymbol{x})$ *ako i samo ako* $\nabla f(\boldsymbol{x}) = 0$. *Trivijalno, ako je* $T(\boldsymbol{x}) = \boldsymbol{x}$ *onda* $f(T(\boldsymbol{x})) = f(\boldsymbol{x})$. *Da bismo pokazali obrnutu implikaciju, pretpostavimo da je* $f(T(\boldsymbol{x})) = f(\boldsymbol{x})$. *Koristeći poslednji lanac nejednakosti (3.9) i pretpostavku, imamo da je* $h(\boldsymbol{x}, \boldsymbol{x}) = f(\boldsymbol{x}) = h(T(\boldsymbol{x}), \boldsymbol{x})$. *Kako je jedinstveni minimizator funkcije* $h(\cdot, \boldsymbol{x})$ *upravo* $T(\boldsymbol{x})$, *sledi* $\boldsymbol{x} = T(\boldsymbol{x})$.

(d) Dokaz sledi nakon jednostavnih alegebarskih transformacija prethodno dokazanog.

Sada možemo da damo osobinu pada niza $\{f(\boldsymbol{x_k})\}_{k \geqslant 0}$ pod pretpostavkom da nijedna iteracija nije usidrena tačka.

Lema 3.5.3. *Neka je* $\{\boldsymbol{x_k}\}_{k \geqslant 0}$ *niz generisan Vajsfeldovom metodom, i pretpostavimo da* $\boldsymbol{x_k} \notin \mathscr{A}$ *za svako* $k \geqslant 0$. *Tada imamo sledeće:*

(a) Niz $\{f(\boldsymbol{x_k})\}_{k \geqslant 0}$ *je nerastući: za svako* $k \geqslant 0$ *važi nejednakost* $f(\boldsymbol{x_{k+1}}) \leqslant f(\boldsymbol{x_k})$.

(b) Za svako k, $f(\boldsymbol{x_{k+1}}) = f(\boldsymbol{x_k})$ *ako i samo ako* $\nabla f(\boldsymbol{x_k}) = 0$.

Dokaz. *(a) Kako* $\boldsymbol{x_k} \notin \mathscr{A}$ *za svako* k, *dokaz sledi ako zamenimo* $\boldsymbol{x} = \boldsymbol{x_k}$ *pod (c) u Lemi 3.4.2.*

(b) Iz Lema 3.4.2 pod (c) imamo da je $f(\boldsymbol{x_k}) = f(\boldsymbol{x_{k+1}}) = f(T(\boldsymbol{x_k}))$ *ako i samo ako* $\boldsymbol{x_k} = \boldsymbol{x_{k+1}} = T(\boldsymbol{x_k})$. *Iz iste Leme pod (d) dobijamo da je poslednje navedeno ekvivalentno sa* $\nabla f(\boldsymbol{x_k}) = 0$.

Naime, pokazali smo da je niz $\{f(\boldsymbol{x_k})\}_{k\geqslant 0}$ striktno opadajući sve dok nismo u stacionarnoj tački. Pretpostavka koju smo koristili, a to je da $\boldsymbol{x_k} \notin \mathscr{A}$, može da predstavlja problem kada treba da proverimo da li je ispunjena. Jedan od načina koji može da nam garantuje da niz generisan Vajsfeldovim metodom ne sadrži usidrene tačke, jeste da izaberemo početnu tačku $\boldsymbol{x_0}$ tako da je vrednost funkcije u toj tački manja od vrednosti funkcije u usidrenim tačkama, to jest:

$$f(\boldsymbol{x_0}) < \min\{f(\boldsymbol{a_1}), f(\boldsymbol{a_2}), ..., f(\boldsymbol{a_m})\}.$$

Kombinovanjem ove pretpostavke sa monotonošću vrednosti funkcije niza generisanog Vajsfeldovim metodom, implicira da iteracije ne uključuju usidrene tačke. U nastavku dajemo dokaz da pod ovom pretpostavkom svaki konvergentan podniz niza $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ konvergira ka stacionarnoj tački.

Teorema 3.5.1. *Neka je $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ niz generisan Vajsfeldovom metodom i pretpostavimo da važi $f(\boldsymbol{x_0}) < min\{f(\boldsymbol{a_1}), f(\boldsymbol{a_2}), ..., f(\boldsymbol{a_m})\}$. Tada je granična vrednost bilo kog konvergentnog podniza $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ stacionarna tačka funkcije f.*

Dokaz. *Neka je $\{\boldsymbol{x_{k_n}}\}_{n\geqslant 0}$ podniz niza $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ koji konvergira ka tački $\boldsymbol{x}^*$. Iz monotonosti metoda i neprekidnosti funkcije cilja imamo*

$$f(\boldsymbol{x}^*) \leqslant f(\boldsymbol{x_0}) < \min\{f(\boldsymbol{a_1}), f(\boldsymbol{a_2}), ..., f(\boldsymbol{a_m})\}.$$

Dakle, $\boldsymbol{x_k} \notin \mathscr{A}$, pa je $\nabla f(\boldsymbol{x}^*)$ definisan. Pokazaćemo da je $\nabla f(\boldsymbol{x}^*) = 0$. Iz neprekidnosti operatora T u tački $\boldsymbol{x}^*$ sledi da niz $\boldsymbol{x_{k_n+1}} = T(\boldsymbol{x_{k_n}}) \longrightarrow T(\boldsymbol{x}^*)$ kada $n \to \infty$. Niz vrednosti funkcija $\{f(\boldsymbol{x_k})\}_{k\geqslant 0}$ je nerastući i ograničen od dole sa 0, te konvergira ka nekoj vrednosti, koju ćemo označiti sa f^*. Očigledno je da oba niza $\{f(\boldsymbol{x_{k_n}})\}_{n\geqslant 0}$ i $\{f(\boldsymbol{x_{k_n+1}})\}_{n\geqslant 0}$ konvergiraju ka f^*. Iz neprekidnosti funkcije f dobijamo da $f(T(\boldsymbol{x}^*)) = f(\boldsymbol{x}^*) = f^*$, što po Lemi 3.4.2 iz (c) i (d) se implicira da je $\nabla f(\boldsymbol{x}^*) = 0$.

Može se pokazati da za Ferma-Veberov problem, stacionarne tačke su optimalna rešenja, te poslednja teorema pokazuje da za sve granične tačke niza generisanog Vajsfeldovom metodom su globalna optimalna rešenja. Ustvari, moguće je pokazati da ceo niz konvergira ka globalnom optimalnom rešenju, no ta analiza prevazilazi obim ovog rada.

3.6 Konvergencija metode najbržeg pada

3.6.1 Lipšic svojstvo gradijenta

U ovom odeljku predstavićemo analizu konvergencije metode najbržeg pada primenjenog na neograničen problem minimizacije

$$\min\{f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^n\}.$$

Pretpostavljamo da ako je funkcija cilja f neprekidno diferencijabilna da je njen gradijent ∇f Lipšic neprekidan na celom $\mathbb{R}^n$, što znači

$$\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \leqslant L\|\boldsymbol{x} - \boldsymbol{y}\| \quad \text{za svako} \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

Primetimo da ako je ∇f Lipšic neprekidna sa konstantnom L, tada je Lipšic neprekidna i sa $\widetilde{L}$, za svako $\widetilde{L} \geqslant L$. Prema tome, u suštini postoji beskonačno mnogo Lipšic konstanti za funkciju sa Lipšic gradijentom. Najčešće smo zaiteresovani za najmanju moguću Lipšic konstantu. Klasu funkcija sa Lipšic gradijentom i Lipšic konstantom L označavamo sa $C_L^{1,1}(\mathbb{R}^n)$ ili samo $C_L^{1,1}$. U slučajevima kada nam je vrednost Lipšic konstante nebitna, klasu ćemo označavati sa $C^{1,1}$. Dajemo primere jednostavnih funkcija klase $C^{1,1}$:

- ***Linearne funkcije*** Za dato $\boldsymbol{a} \in \mathbb{R}^n$, funkcija $f(\boldsymbol{x}) = \boldsymbol{a}^T\boldsymbol{x}$ pripada klasi $C_0^{1,1}$.
- ***Kvadratne funkcije*** Neka je A $n \times n$ simetrična matrica, $\boldsymbol{b} \in \mathbb{R}^n$ i $c \in \mathbb{R}$. Tada funkcija $f(\boldsymbol{x}) = \boldsymbol{x}^T A\boldsymbol{x} + 2\boldsymbol{b}^T\boldsymbol{x} + c$ je funkcija klase $C^{1,1}$.

Da bismo izračunali Lipšicovu konstantu kvadratne funkcije $f(\boldsymbol{x}) = \boldsymbol{x}^T A\boldsymbol{x}+2\boldsymbol{b}^T\boldsymbol{x}+c$, koristimo definiciju Lipšic neprekidnosti:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| = 2\|(A\boldsymbol{x} + \boldsymbol{b}) - (A\boldsymbol{y} - \boldsymbol{b})\| = 2\|A\boldsymbol{x} - A\boldsymbol{y}\| \leqslant 2\|A\|\|\boldsymbol{x} - \boldsymbol{y}\|.$$

Zaključujemo da je $2\|A\|$ Lipšic konstanta od ∇f.
Naredna teorema nam govori da za dva puta neprekidno diferencijabilne funkcije važi da je Lipšic neprekidnost funkcije ekvivalenta ograničenosti Hesijana te iste funkcije.

Teorema 3.6.1. *Neka je f dva puta neprekidno diferencijabilna funkcija na celom $\mathbb{R}^n$. Tada su sledeća tvrđenja ekvivalentna:*

(a) $f \in C_L^{1,1}(\mathbb{R}^n)$.

(b) $\|\nabla^2 f(\boldsymbol{x})\| \leqslant L$ *za svako* $\boldsymbol{x} \in \mathbb{R}^n$.

Dokaz. *(b) $\Rightarrow$ (a) Pretpostavimo da $\|\nabla^2 f(\boldsymbol{x})\| \leqslant L$ za svako $\boldsymbol{x} \in \mathbb{R}^n$. Uvedimo smenu $\boldsymbol{z} = \boldsymbol{x} - t(\boldsymbol{y} - \boldsymbol{x})$. Tada, za svako $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ dobijamo*

$$\begin{aligned}\nabla f(\boldsymbol{y}) &= \nabla f(\boldsymbol{x}) + \int_0^1 \nabla^2 f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))(\boldsymbol{y} - \boldsymbol{x})dt \\ &= \nabla f(\boldsymbol{x}) + \Big(\int_0^1 \nabla^2 f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))dt\Big)(\boldsymbol{y} - \boldsymbol{x}),\end{aligned}$$

odnosno

$$\begin{aligned}\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\| &= \Big\|\Big(\int_0^1 \nabla^2 f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))dt\Big)(\boldsymbol{y} - \boldsymbol{x})\Big\| \\ &\leqslant \Big\|\int_0^1 \nabla^2 f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))dt\Big\|\|\boldsymbol{y} - \boldsymbol{x}\| \\ &\leqslant \Big(\int_0^1 \|\nabla^2 f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))\|dt\Big)\|\boldsymbol{y} - \boldsymbol{x}\| \\ &\leqslant L\|\boldsymbol{y} - \boldsymbol{x}\|,\end{aligned}$$

odakle sledi da $f \in C_L^{1,1}$.

(a) $\Rightarrow$ (b) Pretpostavimo sada da $f \in C_L^{1,1}$. Uvedimo smenu $\boldsymbol{z} = \boldsymbol{x} + t\boldsymbol{d}$. Tada za svako $\boldsymbol{d} \in \mathbb{R}^n$ i $\alpha > 0$ važi

$$\nabla f(\boldsymbol{x} + \alpha\boldsymbol{d}) - \nabla f(\boldsymbol{x}) = \int_0^\alpha \nabla^2 f(\boldsymbol{x} + t\boldsymbol{d})\boldsymbol{d}\ dt.$$

Tada,

$$\Big\|\Big(\int_0^\alpha \nabla^2 f(\boldsymbol{x} + t\boldsymbol{d})dt\Big)\boldsymbol{d}\Big\| = \|\nabla f(\boldsymbol{x} + \alpha\boldsymbol{d}) - \nabla f(\boldsymbol{x})\| \leqslant \alpha L\|\boldsymbol{d}\|.$$

pa važi

$$\lim_{\alpha\to 0^+} \frac{\|\nabla f(\boldsymbol{x} + \alpha\boldsymbol{d}) - \nabla f(\boldsymbol{x})\|}{\alpha} \leqslant \lim_{\alpha\to 0^+} \frac{\alpha L\|\boldsymbol{d}\|}{\alpha},$$

odakle sledi

$$\|\nabla^2 f(\boldsymbol{x})\boldsymbol{d}\| \leqslant L\|\boldsymbol{d}\|,$$

što implicira $\|\nabla^2 f(\boldsymbol{x})\| \leqslant L$.

3.6.2 Lema spusta

Bitna karakteristika funkcija klase $C^{1,1}$ jeste da mogu biti ograničene od gore sa kvadratnom funkcijom na celom prostoru. Taj rezultat poznat je pod nazivom lema spusta i fundamentalna je u dokazivanju konvergencije.

Lema 3.6.1. *(Lema spusta). Neka $f \in C_L^{1,1}(\mathbb{R}^n)$. Tada za svako $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$*

$$f(\boldsymbol{y}) \leqslant f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

Dokaz. *Kako važi*

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) = \int_0^1 \Big(\langle \nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})), \boldsymbol{y} - \boldsymbol{x}\rangle dt.$$

to je

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) = \int_0^1 \Big(\langle \nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})), \boldsymbol{y} - \boldsymbol{x}\rangle + \langle \nabla f(\boldsymbol{x}, \boldsymbol{y} - \boldsymbol{x}\rangle - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle\Big) dt.$$

Dalje je

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \int_0^1 \langle \nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle dt.$$

i odatle možemo izvesti sledeću ocenu

$$\begin{aligned}
|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle| &= \Big| \int_0^1 \langle \nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle dt \Big| \\
&\leqslant \int_0^1 |\langle \nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle| dt \\
&\leqslant \int_0^1 \|\nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x})\| \|\boldsymbol{y} - \boldsymbol{x}\| dt \\
&\leqslant \int_0^1 tL\|\boldsymbol{y} - \boldsymbol{x}\|^2 dt \\
&= \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2.
\end{aligned}$$

Primetimo da smo u dokazu prethodne leme pored gornje dobili i donju granicu, tj. važi

$$f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leqslant f(\boldsymbol{y}) \leqslant f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

3.6.3 Konvergencija

Sada kada smo naveli definiciju Lipšic neprekidnosti kao i lemu spusta, spremni smo da pokažemo konvergenciju metode najbržeg pada za funkcije klase $C^{1,1}$. Naravno, ne možemo da garantujemo konvergenciju ka globalnom optimalnom rešenju, no možemo pokazati konvergenciju ka stacionarnim tačkama u smislu da gradijent teži ka nuli. Ovaj odeljak započinjemo sa dovoljnim padom gradijentog metoda u svakoj iteraciji. Da budemo precizniji, pokazaćemo da u svakoj iteraciji pad vrednosti funkcije je konstanta puta kvadratna norma gradijenta.

Lema 3.6.2. *(Lema dovoljnog pada). Pretpostavimo da je $f \in C_L^{1,1}(\mathbb{R}^n)$. Tada za svako $\boldsymbol{x} \in \mathbb{R}^n$ i $t > 0$*

$$f(\boldsymbol{x}) - f(\boldsymbol{x} - t\nabla f(\boldsymbol{x})) \geqslant t\Big(1 - \frac{Lt}{2}\Big)\|\nabla f(\boldsymbol{x})\|^2. \tag{3.10}$$

Dokaz. *Iz prethodne leme (tzv. leme spusta), imamo*

$$\begin{aligned} f(\boldsymbol{x} - t\nabla f(\boldsymbol{x})) &\leqslant f(\boldsymbol{x}) - t\|\nabla f(\boldsymbol{x})\|^2 + \frac{Lt^2}{2}\|\nabla f(\boldsymbol{x})\|^2 \\ &= f(\boldsymbol{x}) - t\Big(1 - \frac{Lt}{2}\Big)\|\nabla f(\boldsymbol{x})\|^2. \end{aligned}$$

Rezultat sledi nakon jednostavnih elementarnih transformacija izraza.

Podsećanja radi, korak metode najbržeg pada smo birali na tri načina: kao konstantu, tačnim linijskim pretraživanjem ili linijskim pretraživanjem unazad. Sada za cilj imamo da pokažemo da bez obzira na koji način izabrali korak osobina dovoljnog pada će i dalje važiti, što i pokazujemo u nastavku.

Pretpostavimo da je kod konstantnog koraka $t_k = \bar{t} \in (0, \frac{2}{L})$. Zamenjujući $\boldsymbol{x} = \boldsymbol{x_k}$, $\boldsymbol{x} - t\nabla f(\boldsymbol{x}) = \boldsymbol{x_{k+1}}$ i $t = \bar{t}$ u (3.10), imamo nejednakost

$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_{k+1}}) \geqslant \bar{t}\Big(1 - \frac{L\bar{t}}{2}\Big)\|\nabla f(\boldsymbol{x_k})\|^2. \tag{3.11}$$

Primetimo da je zagarantovani pad u metodu najržeg pada po iteraciji

$$\bar{t}\Big(1 - \frac{L\bar{t}}{2}\Big)\|\nabla f(\boldsymbol{x_k})\|^2.$$

Ako želimo da dobijemo najveće garantovano ograničenje pada, tada tražimo maksimum od $g(t) = \bar{t}(1 - \frac{L\bar{t}}{2})$ na $(0, \frac{2}{L})$. Da bismo dobili taj maksimum, tražimo izvod od funkcije g po $\bar{t}$ i izjednačavamo ga sa nulom. Dakle, imamo sledeće:

$$1 - \frac{L\bar{t}}{2} - \bar{t}\frac{L}{2} = 0,$$

odakle sledi

$$\bar{t} = \frac{1}{L},$$

što je očigledno maksimum kvadratne funkcije g i tako dolazimo do poznatog izbora koraka $\frac{1}{L}$. U tom slučaju imamo

$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_{k+1}}) = f(\boldsymbol{x_k}) - f\Big(\boldsymbol{x_k} - \frac{1}{L}\nabla f(\boldsymbol{x_k})\Big) \geqslant \frac{1}{2L}\|\nabla f(\boldsymbol{x_k})\|^2. \tag{3.12}$$

Kod tačnog linijskog pretraživanja, iterativni postupak izgleda

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - t_k\nabla f(\boldsymbol{x_k}),$$

gde je $t_k \in \underset{t\geqslant 0}{\operatorname{argmin}} f(\boldsymbol{x_k} - t_k\nabla f(\boldsymbol{x_k}))$. Po definiciji t_k znamo

$$f(\boldsymbol{x_k} - t\nabla f(\boldsymbol{x_k})) \leqslant f\Big(\boldsymbol{x_k} - \frac{1}{L}\nabla f(\boldsymbol{x_k})\Big),$$

te imamo

$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_k} - t_k\nabla f(\boldsymbol{x_k})) \geqslant f(\boldsymbol{x_k}) - f\Big(\boldsymbol{x_k} - \frac{1}{L}\nabla f(\boldsymbol{x_k})\Big) \geqslant \frac{1}{2L}\|\nabla f(\boldsymbol{x_k})\|^2, \tag{3.13}$$

pri tome je poslednja nejednakost pokazana u (3.12).

Kada je reč o linijskom pretraživanju unazad, tražimo dovoljno mali korak t_k za koji važi

$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_k} - t_k\nabla f(\boldsymbol{x_k})) \geqslant \alpha t_k\|f(\boldsymbol{x_k})\|^2, \tag{3.14}$$

gde je $\alpha \in (0,1)$. Želeli bismo da nađemo donju granicu koraka t_k. Postoje dve opcije. Ili za korak biramo inicijalnu vrednost, tj. $t_k = s$, ili je t_k određen linijskim pretraživanjem unazad, što bi značilo da prethodni korak $\overline{t_k} = \frac{t_k}{\beta}$ nije prihvatljiv i ne zadovoljava (3.14):

$$f(\boldsymbol{x_k}) - f\Big(\boldsymbol{x_k} - \frac{t_k}{\beta}\nabla f(\boldsymbol{x_k})\Big) < \alpha t_k \|f(\boldsymbol{x_k})\|^2. \tag{3.15}$$

Zamenjujući $\boldsymbol{x} = \boldsymbol{x_k}$ i $t = \dfrac{t_k}{\beta}$ u (3.10) dobijamo

$$f(\boldsymbol{x_k}) - f\Big(\boldsymbol{x_k} - \frac{t_k}{\beta}\nabla f(\boldsymbol{x_k})\Big) \geqslant \frac{t_k}{\beta}\Big(1 - \frac{Lt_k}{2\beta}\Big)\|\nabla f(\boldsymbol{x_k})\|^2,$$

što u kombinaciji sa (3.15) implicira

$$\frac{t_k}{\beta}\Big(1 - \frac{Lt_k}{2\beta}\Big) < \alpha\frac{t_k}{\beta},$$

što je isto ako zapišemo kao $t_k > \dfrac{2(1-\alpha)\beta}{L}$. Naime, u linijskom pretraživanju unazad imamo

$$t_k \geqslant \min\Big\{s, \frac{2(1-\alpha)\beta}{L}\Big\},$$

što u kombinaciji sa (3.14) daje

$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_k} - t_k\nabla f(\boldsymbol{x_k})) \geqslant \alpha \min\Big\{s, \frac{2(1-\alpha)\beta}{L}\Big\}\|\nabla f(\boldsymbol{x_k})\|^2. \tag{3.16}$$

Sumiranje navedene tri nejednakosti (3.12), (3.13) i (3.16) formalno dajemo u narednoj teoremi.

Teorema 3.6.2. *Neka $f \in C_L^{1,1}(\mathbb{R}^n)$. Neka je $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ niz generisan metodom najbržeg pada za rešavanje problema*

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

sa jednom od sledećih strategija za korak:

- *konstantnim korakom $\overline{t} \in (0, \frac{2}{L})$,*
- *tačnim linijskim pretraživanjem,*
- *linijskim pretraživanjem unazad sa parametrima $s \in \mathbb{R}^+, \alpha \in (0,1),\ i\ \beta \in (0,1)$.*

Tada,

$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_{k+1}}) \geqslant M\|\nabla f(\boldsymbol{x_k})\|^2, \tag{3.17}$$

gde je

$$M = \begin{cases} \overline{t}\big(1 - \frac{\overline{t}L}{2}\big) & \textit{konstantan korak,} \\ \frac{1}{2L} & \textit{tačno linijsko pretraživanje,} \\ \alpha\min\big\{s, \frac{2(1-\alpha)\beta}{L}\big\} & \textit{linijsko pretraživanje unazad.} \end{cases}$$

Sada pokazujemo da norma gradijenata $\|\nabla f(\boldsymbol{x_k})\|^2$ konvergira ka nuli.

Teorema 3.6.3. *Neka $f \in C_L^{1,1}(\mathbb{R}^n)$. Neka je $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ niz generisan metodom najbržeg pada za rešavanje problema*

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

sa jednom od sledećih strategija za korak:

- *konstantnim korakom* $\bar{t} \in (0, \frac{2}{L})$,
- *tačnim linijskim pretraživanjem,*
- *linijskim pretraživanjem unazad sa parametrima* $s \in \mathbb{R}^+, \alpha \in (0,1)$, *i* $\beta \in (0,1)$.

Neka je f *ograničena odole na celom* $\mathbb{R}^n$, *to jest, postoji* $m \in \mathbb{R}$ *tako da je* $f(\boldsymbol{x}) > 0$ *za svako* $\boldsymbol{x} \in \mathbb{R}^n$. *Tada važi:*

(a) Niz $\{f(\boldsymbol{x_k})\}_{k\geqslant 0}$ *je nerastući. Štaviše, za svako* $k \geqslant 0$, $f(\boldsymbol{x_{k+1}}) < f(\boldsymbol{x_k})$ *osim ako je* $\nabla f(\boldsymbol{x_k}) = 0$.

(b) $\nabla f(\boldsymbol{x_k}) \to 0$, *kad* $k \to \infty$.

Dokaz. *(a) Iz* (3.17) *imamo*

$$f(\boldsymbol{x_k}) - f(\boldsymbol{x_{k+1}}) \geqslant M\|\nabla f(\boldsymbol{x_k})\|^2 \geqslant 0$$

za neku konstantu $M > 0$, *te* $f(\boldsymbol{x_k}) = f(\boldsymbol{x_{k+1}})$ *važi samo ako* $\nabla f(\boldsymbol{x_k}) = 0$.
(b) Kako je niz $\{f(\boldsymbol{x_k})\}_{k\geqslant 0}$ *nerastući i ograničen odole, samim tim konvergira. Tako da,* $f(\boldsymbol{x_k}) - f(\boldsymbol{x_{k+1}}) \to 0$ *kad* $k \to 0$, *što u kombinaciji sa* (3.17) *implicira da* $\|\nabla f(\boldsymbol{x_k})\| \to 0$ *kad* $k \to 0$.

I napokon, u narednoj teoremi dajemo brzinu konvergencije, tačnije ocenu norme gradijenta.

Teorema 3.6.4. *Neka važe pretpostavke Teoreme 3.5.2. i neka je* f^* *granica konvergentnog niza* $\{f(\boldsymbol{x_k})\}_{k\geqslant 0}$. *Tada za svako* $n = 0, 1, 2...$ *važi*

$$\min_{k=0,1,2...,n} \|\nabla f(\boldsymbol{x_k})\| \leqslant \sqrt{\frac{f(\boldsymbol{x_0}) - f^*}{M(n+1)}},$$

gde je

$$M = \begin{cases} \bar{t}\left(1 - \frac{\bar{t}L}{2}\right) & \text{konstantan korak,} \\ \frac{1}{2L} & \text{tačno linijsko pretraživanje,} \\ \alpha \min\left\{s, \frac{2(1-\alpha)\beta}{L}\right\} & \text{linijsko pretraživanje unazad.} \end{cases}$$

Dokaz. *Sumiranjem* (3.17) *po* $i = 0, 1, 2...$ *dobijamo*

$$f(\boldsymbol{x_0}) - f(\boldsymbol{x_{n+1}}) \geqslant M \sum_{k=0}^{n} \|\nabla f(\boldsymbol{x_k})\|^2.$$

Kako je $f(\boldsymbol{x_{n+1}}) \geqslant f^*$, *zaključujemo*

$$f(\boldsymbol{x_0}) - f^* \geqslant M \sum_{k=0}^{n} \|\nabla f(\boldsymbol{x_k})\|^2.$$

Koristeći poslednju nejednakost zajedno sa činjenicom da za svako $k = 0, 1, 2...$ *važi očigledna nejednakost* $\|\nabla f(\boldsymbol{x_k})\|^2 \geqslant \min_{k=0,1,...,n} \|\nabla f(\boldsymbol{x_k})\|^2$, *sledi*

$$f(\boldsymbol{x_0}) - f^* \geqslant M(n+1) \min_{k=0,1,...,n} \|\nabla f(\boldsymbol{x_k})\|^2,$$

odakle lako uočavamo traženi rezultat.

4 Metode konjugovanih gradijenata

Metode konjugovanih gradijenata su pogodne za rešavanje linearnih sistema, kao i za rešavanje problema nelinearne optimizacije bez ograničenja. Postupak konjugovanih gradijenata prvi put je predstavljen u radu [10] u kojem je posmatran problem rešavanja simetričnog, pozitivno definitnog linearnog sistema. U slučaju konveksne, kvadratne funkcije, poznato je da rešavanje problema minimizacije je ekvivalentno rešenju sistema linearnih jednačina sa simetričnom, pozitivno definitnom matricom. Ta ideja je kasnije poslužila za razvijanjem postupaka konjugovanih gradijenata za rešavanje problema nelinearne optimizacije bez ograničenja. Prvi metod su predložili Flečer i Rivs u svom radu [7] pretpostavljajući da se u okolini minimuma problema funkcija cilja aproksimira konveksnom, kvadratnom funkcijom i predstavili postupak koji u konačno mnogo koraka daje rešenje. Nakon njihovog objavljivanja rada, postupci konjugovanih gradijenata za rešavanje nelinearne optimizacije bez ograničenja postaju veoma atraktivni i rapidno se razvijaju sve do danas.

Relevanta literatura ovog poglavlja jeste [6], [13], [15] i [17].

4.1 Linearni metod konjugovanih gradijenata

Linearni metod konjugovanih gradijenata je iterativni postupak za rešavanje sistema linearnih jednačina oblika

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{4.1}$$

gde je $\boldsymbol{A}$ $n \times n$ simetrična pozitivno definitna matrica, i vektor $\boldsymbol{b} \in \mathbb{R}^n$. Ako definišemo funkciju $f : \mathbb{R}^n \to \mathbb{R}$ na sledeći način:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x},$$

gde je $\boldsymbol{A}$ $n \times n$ simetrična pozitivno definitna matrica, i vektor $\boldsymbol{b} \in \mathbb{R}^n$, tada je problem (4.1) ekvivalentan problemu minimizacije

$$\min f(\boldsymbol{x}). \tag{4.2}$$

Drugim rečima, oba problema (4.1) i (4.2) imaju isto jedinstveno rešenje. Ova ekvivalencija nam dozvoljava interpretaciju metoda konjugovanih gradijenata kao algoritam za rešavanje linearnih sistema ili kao tehnika za minimizaciju konveksnih kvadratnih funkcija. Primetimo da je gradijent funkcije f jednak rezidualu linearnog sistema, tj.

$$\nabla f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{r}(\boldsymbol{x}). \tag{4.3}$$

4.1.1 Konjugovani vektori

Metod konjugovanih gradijenata poznat je po osobini formiranja skupa vektora (pravaca) koji zadovoljavaju svojstvo konjugovanosti, stoga ovu sekciju posvećujemo upravo konjugovanim vektorima.

Naime, poznato je iz literature da bitan faktor za efikasnost metoda iterativnog tipa za rešavanje problema minimizacije zadate funkcije jeste način na koji određujemo pravac pretrage u svakoj iteraciji. Može se pokazati da za kvadratnu funkciju (4.2) najbolji pravac pretrage upravo vektor iz skupa $\boldsymbol{A}$-*konjugovanih pravaca.* Za vektore (pravce) $\boldsymbol{d_1}$ i $\boldsymbol{d_2}$ iz $\mathbb{R}^n$ kažemo da su $\boldsymbol{A}$-konjugovani ukoliko je $\boldsymbol{d_1}^T A\boldsymbol{d_2} = 0$, gde je $\boldsymbol{A}$ matrica kvadratne funkcije f posmatranog problema (4.2). Preciznije, imamo sledeću definiciju.

Definicija 4.1.1. Neka je $\boldsymbol{A}$ realna, simetrična kvadratna matrica reda n. Za pravce $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k} \in \mathbb{R}^n$ kažemo da su $\boldsymbol{A}$-konjugovani ako važi

$$\boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_j} = 0, \text{ za svako } i \neq j.$$

Sledeća lema govori o linearnoj nezavisnosti $\boldsymbol{A}$-konjugovanih pravaca u slučaju da je matrica $\boldsymbol{A}$ pozitivno definitna.

Lema 4.1.1. *Neka je $\boldsymbol{A}$ simetrična, pozitivno definitna matrica reda n. Ukoliko su nenula pravci $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k} \in \mathbb{R}^n$, $k \leqslant n-1$, $\boldsymbol{A}$-konjugovani, tada su i linearno nezavisni.*

Dokaz. *Neka su $\alpha_0, \alpha_1, ..., \alpha_k$ skalari takvi da je linearna kombinacija vektora $\boldsymbol{d_i}, i = 0, 1, ..., k$ jednaka nuli, odnosno*

$$\alpha_0 \boldsymbol{d_0} + \alpha_1 \boldsymbol{d_1} + ... + \alpha_k \boldsymbol{d_k} = 0.$$

Ukoliko pomnožimo ovu linearnu kombinaciju sa $\boldsymbol{d_j}^T \boldsymbol{A}$ sa leve strane za svako $0 \leqslant j \leqslant k$, i imajući u vidu da su pravci $\boldsymbol{A}$-konjugovani, dobijamo

$$\alpha_j \boldsymbol{d_j}^T \boldsymbol{A}\boldsymbol{d_j} = 0,$$

jer su svi ostali članovi $\boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_j} = 0$, za svako $i \neq j$. Kako je, prema pretpostavci leme, matrica $\boldsymbol{A}$ pozitivno definitna i vektori $\boldsymbol{d_i} \neq 0$, sledi da je

$$\alpha_j = 0, \ \boldsymbol{za\ svako}\ j = 0, 1, ..., k,$$

što po definiciji znači da su vektori $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}$, $k \leqslant n-1$ linearno nezavisni.

Primetimo da u slučaju $\boldsymbol{A} = \boldsymbol{I}$, gde je $\boldsymbol{I}$ jedinična matrica odgovarajućeg reda, $\boldsymbol{A}$-konjugovani vektori $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}$ su zapravo ortogonalni vektori, odnosno važi

$$\boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_j} = \boldsymbol{d_i}^T \boldsymbol{d_j} = 0, \text{ za svako } i \neq j.$$

Dakle, za $\boldsymbol{A} = \boldsymbol{I}$, svojstvo $\boldsymbol{A}$-konjugovanosti skupa vektora se svodi na svojstvo ortogonalnosti.
Lema 4.1.1 koju smo dokazali ima značajnu ulogu pri minimizaciji problema (4.2). Pretpostavimo da su vektori $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}$, $k \leqslant n-1$, međusobno $\boldsymbol{A}$-konjugovani i posmatramo vektor $\boldsymbol{y}$, koji predstavlja njihovu linearnu kombinaciju

$$\boldsymbol{y} = \sum_{i=0}^{k} \alpha_i \boldsymbol{d_i}.$$

Odredimo vrednost funkcije f u tački $\boldsymbol{y}$:

$$\begin{aligned} f(\boldsymbol{y}) &= f(\sum_{i=0}^{k} \alpha_i \boldsymbol{d_i}) \\ &= \frac{1}{2}(\sum_{i=0}^{k} \alpha_i \boldsymbol{d_i})^T \boldsymbol{A}(\sum_{i=0}^{k} \alpha_i \boldsymbol{d_i}) - \boldsymbol{b}^T(\sum_{i=0}^{k} \alpha_i \boldsymbol{d_i}) \\ &= \frac{1}{2}\sum_{i=0}^{k}\sum_{j=0}^{k} \alpha_i \alpha_j \boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_j} - \sum_{i=0}^{k} \alpha_i \boldsymbol{b}^T \boldsymbol{d_i} \\ &= \sum_{i=0}^{k}(\frac{1}{2}\alpha_i^2 \boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_i} - \alpha_i \boldsymbol{b}^T \boldsymbol{d_i}). \end{aligned}$$

Sada lako možemo minimizovati funkciju f po svim vektorima $\boldsymbol{y}$ koji su linearna kombinacija $\boldsymbol{A}$-konjugovanih vektora $\boldsymbol{d_i}$, $i = 0, 1, ..., k$. Imamo da je

$$\begin{aligned}\min_{\boldsymbol{y}} f(\boldsymbol{y}) &= \min_{\alpha_i \in \mathbb{R}} f(\sum_{i=0}^{k} \alpha_i \boldsymbol{d_i}) \\ &= \min_{\alpha_i \in \mathbb{R}} \sum_{i=0}^{k} (\frac{1}{2}\alpha_i^2 \boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_i} - \alpha_i \boldsymbol{b}^T \boldsymbol{d_i}) \\ &= \sum_{i=0}^{k} \min_{\alpha_i \in \mathbb{R}} (\frac{1}{2}\alpha_i^2 \boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_i} - \alpha_i \boldsymbol{b}^T \boldsymbol{d_i}).\end{aligned}$$

Primetimo da promenljiva $\boldsymbol{y}$ zavisi od $k+1$ promenljive, to jest od α_i, za $i = 0, 1, ..., k$. Dakle, f će biti minimalno kada svaka od te $k + 1$ vrednosti bude minimalna sama za sebe. Iz tog razloga poslednja jednakost prethodno izvedenog izraza važi.

Prema tome, vidimo da se polazni problem (4.2) svodi na sumu $k + 1$ jednodimenzionalnih problema minimizacije. Svaki od ovih problema se može rešiti izjednačavanjem izvoda po α_i sa nulom

$$\alpha_i(\boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_i}) - \boldsymbol{b}^T \boldsymbol{d_i} = 0, \quad i = 0, 1, ..., k,$$

odakle je

$$\alpha_i = \frac{\boldsymbol{b}^T \boldsymbol{d_i}}{\boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_i}}, \quad i = 0, 1, ..., k.$$

Kada smo odredili koeficijente α_i, sada dobijamo konačno rešenje

$$\boldsymbol{x}^* = \boldsymbol{y} = \sum_{i=0}^{k} \frac{\boldsymbol{b}^T \boldsymbol{d_i}}{\boldsymbol{d_i}^T \boldsymbol{A}\boldsymbol{d_i}} \boldsymbol{d_i}.$$

Ukoliko je $k = n - 1$, tada svaki vektor $\boldsymbol{y} \in \mathbb{R}^n$ može biti predstavljen kao linearna kombinacija vektora $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_{n-1}}$, s obzirom da su linearno nezavisni i da ih ima tačno n.

Važnost konjugovanosti leži u tome što možemo minimizirati funkciju $f(\cdot)$ u n koraka uzastopnim minimiziranjem duž pojedinih pravaca iz skupa konjugovanih vektora. Da bismo verifikovali ovu činjenicu, posmatraćemo *metod konjugovanih pravaca.*

No, pre toga dajemo formulu za dužinu koraka α_k koju koristimo dalje u radu u algoritmima. Naime, iz [13] znamo da za datu početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ i skup konjugovanih pravaca $\boldsymbol{d_0}, ..., \boldsymbol{d_{n-1}}$, i niz $\{\boldsymbol{x_k}\}$ generisan

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \alpha_k \boldsymbol{d_k},$$

gde je α_k jednodimenzionalni minimizator kvadratne funkcije f duž pravca $\boldsymbol{x_k} + \alpha \boldsymbol{d_k}$, eksplicitni izraz za α_k ima oblik:

$$\alpha_k = -\frac{\boldsymbol{r_k}^T \boldsymbol{d_k}}{\boldsymbol{k}^T \boldsymbol{A}\boldsymbol{d_k}}.$$

4.1.2 Metod konjugovanih pravaca

I dalje posmatramo funkciju $f : \mathbb{R}^n \to \mathbb{R}$ definisanu u prethodnoj sekciji kao $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}$, gde je $\boldsymbol{A}$ simetrična, pozitivno definitna matrica reda n i $\boldsymbol{b} \in \mathbb{R}^n$

dati vektor. U nastavku izlažemo metodu konjugovanih pravaca za rešavanje našeg problema (4.2). Kako je $\boldsymbol{A} > 0$, tada funkcija ima globalni minimum koji se može naći rešavanjem jednačine $\boldsymbol{Ax} = \boldsymbol{b}$. Iz tog razloga, metod konjugovanih pravaca se može koristiti i za rešavanje sistema linearnih jednačina sa pozitivno definitnom matricom, odnosno, za rešavanje problema (4.1).

Suština osnovne varijante metode konjugovanih pravaca za minimizaciju kvadratne funkcije je konstrukcija iterativnog niza $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ koristeći skup $\boldsymbol{A}$-konjugovanih pravaca. Ukoliko nam je poznat skup $\boldsymbol{A}$-konjugovanih vektora $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_{n-1}}$ kao i početna tačka $\boldsymbol{x_0}$, osnovna metoda konjugovanih pravaca je relativno jednostavna i sastoji se iz sledećeg niza koraka.

Algoritam : Metod konjugovanih pravaca

***Korak* 0.** Izabrati dopustivu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$.
***Korak* 1.** Izračunati $\nabla f(\boldsymbol{x_k}) = \boldsymbol{Ax_k} - \boldsymbol{b} = \boldsymbol{r_k}$.
***Korak* 2.** Odrediti $\alpha_k = -\dfrac{\boldsymbol{r_k d_k}}{\boldsymbol{d_k}^T \boldsymbol{A d_k}}$.
***Korak* 3.** Odrediti narednu tačku iterativnog postupka po formuli

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \alpha_k \boldsymbol{d_k}.$$

***Korak* 4.** Ispitati da li za $\boldsymbol{x_k}$ važi izabrani kriterijum zaustavljanja. Ukoliko važi, iterativni postupak se zaustavlja i uzimamo da je $\boldsymbol{x_{k+1}} \approx \boldsymbol{x}^*$. U suprotnom, postaviti da je $k = k + 1$ i ići na Korak 1.

Teorema 4.1.1. *Za svako $\boldsymbol{x_0} \in \mathbb{R}^n$ niz $\{\boldsymbol{x_k}\}_{k\leqslant 0}$ generisam metodom konjugovanih pravaca konvergira ka $\boldsymbol{x}^*$ linearnog sistema* (4.1) *u najviše n koraka.*

Dokaz. *Neka su $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_{n-1}}$ zadati skup $\boldsymbol{A}$-konjugovanih pravaca i $\boldsymbol{x_0}$ proizvoljno izabrana tačka. Posmatramo vektor $\boldsymbol{x}^* - \boldsymbol{x_0}$. Kako su vektori $\boldsymbol{d_i} \in \mathbb{R}^n$, $i = 0, 1, ..., n-1$ linearno nezavisni, postoje konstante β_i, $i = 0, 1, ..., n-1$ takve da važi*

$$\boldsymbol{x}^* - \boldsymbol{x_0} = \beta_0 \boldsymbol{d_0} + \beta_1 \boldsymbol{d_1} + ... + \beta_{n-1} \boldsymbol{d_{n-1}}.$$

Množeći obe strane sa $\boldsymbol{d_k}^T \boldsymbol{A}$, za $0 \leqslant k < n$, dobijamo

$$\boldsymbol{d_k}^T \boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x_0}) = \beta_k \boldsymbol{d_k}^T \boldsymbol{A d_k},$$

jer su vektori $\boldsymbol{d_k}$ i $\boldsymbol{d_i}$ A-konjugovani za $k \neq i$, te važi $\boldsymbol{d_k}^T \boldsymbol{A d_i} = 0$ za $k \neq i$. Iz poslednje jednakosti sledi

$$\beta_k = \frac{\boldsymbol{d_k}^T A(\boldsymbol{x}^* - \boldsymbol{x_0})}{\boldsymbol{d_k}^T A \boldsymbol{d_k}}.$$

Sa druge strane, svaku tačku iterativnog niza $\{\boldsymbol{x_k}\}_{k\geqslant 1}$ možemo predstaviti u obliku

$$\boldsymbol{x_k} = \boldsymbol{x_{k-1}} + \alpha_{k-1} \boldsymbol{d_{k-1}} = \boldsymbol{x_0} + \alpha_0 \boldsymbol{d_0} + \alpha_1 \boldsymbol{d_1} + ... + \alpha_{k-1} \boldsymbol{d_{k-1}},$$

odakle je

$$\boldsymbol{x_k} - \boldsymbol{x_0} = \alpha_0 \boldsymbol{d_0} + \alpha_1 \boldsymbol{d_1} + ... + \alpha_{k-1} \boldsymbol{d_{k-1}}.$$

Vektor $\boldsymbol{x}^* - \boldsymbol{x_0}$ možemo predstaviti i na sledeći način

$$\boldsymbol{x}^* - \boldsymbol{x_0} = (\boldsymbol{x}^* - \boldsymbol{x_k}) + (\boldsymbol{x_k} - \boldsymbol{x_0}).$$

Množenjem gornje jednakosti vektorom $\boldsymbol{d_k}^T \boldsymbol{A}$ sa obe strane, dobijamo

$$\boldsymbol{d_k}^T \boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x_0}) = \boldsymbol{d_k}^T \boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x_k}) + \boldsymbol{d_k}^T \boldsymbol{A}(\boldsymbol{x_k} - \boldsymbol{x_0}),$$

odnosno,

$$\boldsymbol{d_k}^T\boldsymbol{A}(\boldsymbol{x}^*-\boldsymbol{x_0})=\boldsymbol{d_k}^T\boldsymbol{A}(\boldsymbol{x}^*-\boldsymbol{x_k})+\boldsymbol{d_k}^T\boldsymbol{A}(\boldsymbol{x_k}-\boldsymbol{x}^*)+\boldsymbol{d_k}^T\boldsymbol{A}(\boldsymbol{x}^*-\boldsymbol{x_0}),$$

odakle je

$$0=\boldsymbol{d_k}^T\boldsymbol{A}(\boldsymbol{x}^*-\boldsymbol{x_k})+\boldsymbol{d_k}^T(\boldsymbol{A}\boldsymbol{x_k}-\boldsymbol{A}\boldsymbol{x}^*).$$

S obzirom da važi $\nabla f(\boldsymbol{x_k})=\boldsymbol{A}\boldsymbol{x_k}-\boldsymbol{b}$, $\boldsymbol{A}\boldsymbol{x}^*=\boldsymbol{b}$ *i vektor* $\boldsymbol{d_k}$ *je* $\boldsymbol{A}$*-konjugovan sa* $\boldsymbol{d_0},\boldsymbol{d_1},...,\boldsymbol{d_{k-1}}$*, poslednji izraz se svodi na*

$$\boldsymbol{d_k}^T\boldsymbol{A}(\boldsymbol{x}^*-\boldsymbol{x_0})=\boldsymbol{d_k}^T\boldsymbol{A}(\boldsymbol{x}^*-\boldsymbol{x_k})=-\boldsymbol{d_k}^T\nabla f(\boldsymbol{x_k}).$$

Uvrštavanjem $\boldsymbol{d_k}^T\boldsymbol{A}(\boldsymbol{x}^*-\boldsymbol{x_0})=-\boldsymbol{d_k}^T\nabla f(\boldsymbol{x_k})$ *u izraz za* β_k *koji smo prethodno izveli, dobijamo*

$$\beta_k=-\frac{\boldsymbol{d_k}^T\nabla f(\boldsymbol{x_k})}{\boldsymbol{d_k}^T\boldsymbol{A}\boldsymbol{d_k}}=\alpha_k.$$

U slučaju $k=n-1$*, imamo da je* $\beta_k=\alpha_k$*,* $k=0,1,...,n-1$*, te je* $\boldsymbol{x_n}=\boldsymbol{x}^*$*, što je i trebalo dokazati.*

Teorema 4.1.2. *Neka je* $\boldsymbol{x_0}\in\mathbb{R}^n$ *proizvoljna početna tačka i pretpostavimo da je niz* $\{\boldsymbol{x_k}\}_{k\geqslant 0}$ *generisam metodom konjugovanih pravaca. Tada je*

$$\boldsymbol{r_k}^T\boldsymbol{d_i}=0,\qquad za\ i=0,1,...,k-1, \tag{4.4}$$

i $\boldsymbol{x_k}$ *je minimizator funkcije* $f(\boldsymbol{x})=\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}^T\boldsymbol{x}$ *na skupu*

$$\{\boldsymbol{x}|\boldsymbol{x}=\boldsymbol{x_0}+span\{\boldsymbol{d_0},\boldsymbol{d_1},...,\boldsymbol{d_{k-1}}\}\}, \tag{4.5}$$

(pri tome pod span$\{\boldsymbol{d_0},\boldsymbol{d_1},...,\boldsymbol{d_{k-1}}\}$ *podrazumevamo vektorski potprostor generisan tim vektorima).*

Dokaz. *Prvo dokazujemo da tačka* $\widetilde{x}$ *minimizira* f *na skupu (4.5) ako i samo ako* $\boldsymbol{r}(\widetilde{\boldsymbol{x}})^T\boldsymbol{d_i}=0$ *za svako* $i=0,1,...,k-1$*.. Definišemo funkciju* $h(\sigma)=f(\boldsymbol{x_0}+\sigma_0\boldsymbol{d_0}+\sigma_1\boldsymbol{d_1}+...+\sigma_{k-1}\boldsymbol{d_{k-1}})$*, gde je* $\sigma=(\sigma_0,\sigma_1,...,\sigma_{k-1})^T$*. Kako je* $h(\sigma)$ *striktno konvekna, kvadratna funkcija, postoji jedinstveni minimizator* σ^* *koji zadovoljava*

$$\frac{\partial h(\sigma^*)}{\partial\sigma_i}=0,\ \ za\ svako\ i=0,1,...,k-1.$$

Koristeći definiciju izvoda složene funkcije, poslednja jednakost implicira

$$\nabla f(\boldsymbol{x_0}+\sigma_0^*\boldsymbol{d_0}+\sigma_1^*\boldsymbol{d_1}+...+\sigma_{k-1}^*\boldsymbol{d_{k-1}})^T\boldsymbol{d_i}=0,\ \ i=0,1,...,k-1.$$

Pozivajući se na formulu (4.3)*, imamo za minimizator* $\widetilde{\boldsymbol{x}}=\boldsymbol{x_0}+\sigma_0\boldsymbol{d_0}+\sigma_1\boldsymbol{d_1}+...+\sigma_{k-1}\boldsymbol{d_{k-1}}$ *na skupu (4.5) da zadovoljava* $\boldsymbol{r}(\widetilde{\boldsymbol{x}})^T\boldsymbol{d_i}=0$*, kao što se i tvrdi.*

Sada indukcijom pokazujemo da $\boldsymbol{x_k}$ *zadovoljava (4.4). Za* $k=1$*, imamo iz činjenice da* $\boldsymbol{x_1}=\boldsymbol{x_0}+\alpha_0\boldsymbol{d_0}$ *minimizira* f *duž* $\boldsymbol{d_0}$ *sledi* $\boldsymbol{r_1}^T\boldsymbol{d_1}=0$*. Pretpostavimo sada da važi indukcijska hipoteza, naime* $\boldsymbol{r_{k-1}}^T\boldsymbol{d_i}=0$*, za* $i=0,1,...,k-2$*. Znamo da važi*

$$\boldsymbol{r_k}=\boldsymbol{r_{k-1}}+\alpha_{k-1}\boldsymbol{A}\boldsymbol{d_{k-1}},$$

tako da ako pomnožimo poslednju jednakost sa $\boldsymbol{d_{k-1}}$ *dobijamo*

$$\boldsymbol{d_{k-1}}^T\boldsymbol{r_k}=\boldsymbol{d_{k-1}}^T\boldsymbol{r_{k-1}}+\alpha_{k-1}\boldsymbol{d_{k-1}}^T\boldsymbol{A}\boldsymbol{d_{k-1}}.$$

Zamenjujući α_k *po definicija, tj.* $\alpha_k = -\dfrac{\boldsymbol{r_k}^T \boldsymbol{d_k}}{\boldsymbol{d_k}^T \boldsymbol{A} \boldsymbol{d_k}}$ *dobijamo:*

$$\boldsymbol{d_{k-1}}^T \boldsymbol{r_k} = \boldsymbol{d_{k-1}}^T \boldsymbol{r_{k-1}} - \frac{\boldsymbol{r_k}^T \boldsymbol{d_k}}{\boldsymbol{d_k}^T \boldsymbol{A} \boldsymbol{d_k}} \boldsymbol{d_{k-1}}^T \boldsymbol{A} \boldsymbol{d_{k-1}} = 0.$$

Međutim, za ostale vektore $\boldsymbol{d_i}$, $i = 0, 1, ..., k-2$, *imamo*

$$\boldsymbol{d_i}^T \boldsymbol{r_k} = \boldsymbol{d_i}^T \boldsymbol{r_{k-1}} + \alpha_{k-1} \boldsymbol{d_i}^T \boldsymbol{A} \boldsymbol{d_i} = 0,$$

gde je $\boldsymbol{d_i}^T \boldsymbol{r_{k-1}} = 0$ *po indukcijskoj hipotezi i* $\boldsymbol{d_i}^T \boldsymbol{A} \boldsymbol{d_i} = 0$ *zbog konjugovanosti vektora* $\boldsymbol{d_i}$. *Dakle, pokazali smo da je* $\boldsymbol{r_k}^T \boldsymbol{d_i} = 0$, *za* $i = 0, 1, ..., k-1$, *te je naš dokaz završen.*

Kod metoda konjugovanih pravaca možemo primetiti da je neophodno zadati ne samo početnu tačku $\boldsymbol{x_0}$, već i skup $\boldsymbol{A}$-konjugovanih vektora (pravaca). Međutim, postoji varijanta metoda konjugovanih pravaca pod nazivom *metoda konjugovanih gradijenata*, koja tokom iterativnog procesa formira skup $\boldsymbol{A}$-konjugovanih vektora.

4.1.3 Metod konjugovanih gradijenata – osnovne osobine

Metod konjugovanih gradijenata je metod konjugovanih pravaca sa posebnom osobinom. Generisanjem njegovog skupa konjugovanih vektora, vektor $\boldsymbol{d_k}$ se računa korišćenjem samo prethodnog vektora $\boldsymbol{d_{k-1}}$. Nije neophodno poznavanje prethodnih vektora $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_{k-2}}$, a vektor $\boldsymbol{d_k}$ je implicitno konjugovanim sa njima. Ovo svojstvo vredno pažnje govori to da metod zahteva malo skladištenja memorije i računskih operacija.

Kod metoda konjugovanih gradijenata, svaki pravac $\boldsymbol{d_k}$ izabran je da bude linearna kombinacija negativnog reziduala $-\boldsymbol{r_k}$ (što po svojoj definiciji predstavlja pravac najbržeg pada funkcije f), i prethodnog pravca $\boldsymbol{d_{k-1}}$. Zapisujemo

$$\boldsymbol{d_k} = -\boldsymbol{r_k} + \beta_k \boldsymbol{d_{k-1}},$$

gde je skalar β_k određen tako da $\boldsymbol{d_{k-1}}$ i $\boldsymbol{d_k}$ moraju biti konjugovani vektori u odnosu na matricu $\boldsymbol{A}$. Ako poslednju jednakost pomnoženjem sa leve strane sa $\boldsymbol{d_{k-1}}^T \boldsymbol{A}$ dobijamo:

$$\boldsymbol{d_{k-1}}^T \boldsymbol{A} \boldsymbol{d_k} = -\boldsymbol{r_k^T} \boldsymbol{A} \boldsymbol{d_{k-1}} + \beta_k \boldsymbol{d_{k-1}}^T \boldsymbol{A} \boldsymbol{d_{k-1}}.$$

Ako iskoristimo činjenicu da je $\boldsymbol{d_{k-1}}^T \boldsymbol{A} \boldsymbol{d_k} = 0$, imamo

$$\beta_k = \frac{\boldsymbol{r_k^T} \boldsymbol{A} \boldsymbol{d_{k-1}}}{\boldsymbol{d_{k-1}^T} \boldsymbol{A} \boldsymbol{d_{k-1}}}.$$

Izabrali smo da je prvi pravac pretraživanja $\boldsymbol{d_0}$ pravac najbržeg pada kod inicijalne tačke $\boldsymbol{x_0}$. Kao i u opštem metodu konjugovanih pravaca, izvodimo uzastopno jedno-dimenzionalno minimiziranje duž svakog vektora pretraživanja. Formalni oblik algoritma dajemo u nastavku.

Algoritam : Metod konjugovanih gradijenata

Korak 0. Izabrati početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$.

Korak 1. Izračunati $\boldsymbol{r_0} = \boldsymbol{A x_0} - \boldsymbol{b}$. Ukoliko je $\boldsymbol{r_0} = 0$, ili je ispunjen neki drugi kriterijum zaustavljanja, algoritam se zaustavlja i $\boldsymbol{x_0}$ je (približno) rešenje problema. U suprotnom, uzati za $\boldsymbol{d_0} = -\boldsymbol{r_0}$ i ići na Korak 2.

***Korak* 2.** Izračunati $\alpha_k = -\dfrac{\boldsymbol{r_k^T d_k}}{\boldsymbol{d_k^T A d_k}}$.

***Korak* 3.** Odrediti narednu tačku iterativnog niza $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \alpha_k \boldsymbol{d_k}$.

***Korak* 4.** Izračunati $\boldsymbol{r_{k+1}} = \boldsymbol{A x_{k+1}} - \boldsymbol{b}$. Ukoliko je $\boldsymbol{r_{k+1}} = 0$ ili je ispunjen neki drugi kriterijum zaustavljanja, algoritam se zaustavlja i tačka $\boldsymbol{x_{k+1}}$ je (približno) rešenje problema. U suprotnom ići na Korak 5.

***Korak* 5.** Izračunati $\beta_k = \dfrac{\boldsymbol{r_{k+1}^T A d_k}}{\boldsymbol{d_k^T A d_k}}$.

***Korak* 6.** Odrediti $(k+1)$-vi konjugovani vektor $\boldsymbol{d_{k+1}} = -\boldsymbol{r_{k+1}} + \beta_k \boldsymbol{d_k}$. Postaviti $k = k+1$ i ići na Korak 2.

Ova verzija je korisna za proučavanje osnovnih svojstava metode konjugovanih gradijenata, no efikasniju verziju predstavljamo nešto kasnije u našem radu. Prvo pokazujemo jeste da su pravci $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_{n-1}}$ zaista konjugovani, što po Teoremi 4.1.1 implicira završavanje algoritma u n koraka. Teorema koja sledi utvrđuje ovu osobinu kao i druge dve bitne osobine. Prva, reziduali $\boldsymbol{r_k}$ su međusobno ortogonalni. Druga, svaki pravac pretraživanja $\boldsymbol{d_k}$ i rezidual $\boldsymbol{r_k}$ se sadrže u *Krilovom potprostoru stepena* k za $\boldsymbol{r_0}$ definisano kao

$$\mathscr{K}(\boldsymbol{r_0}; k) = span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k \boldsymbol{r_0}\}.$$

Teorema 4.1.3. *Pretpostavimo da k-ta iteracija generisana metodom konjugovanih gradijenata nije rešenje $\boldsymbol{x}^*$. Tada važe sledeća svojstva:*

(a) $\boldsymbol{r_k}^T \boldsymbol{r_i} = 0$, *za* $i = 0, 1, ..., k-1$

(b) $span\{\boldsymbol{r_0}, \boldsymbol{r_1}, ..., \boldsymbol{r_k}\} = span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k \boldsymbol{r_0}\}$

(c) $span\{\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}\} = span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k \boldsymbol{r_0}\}$

(d) $\boldsymbol{d_k}^T \boldsymbol{A d_i} = 0$, *za* $i = 0, 1, ..., k-1$.

Dakle, niz $\{\boldsymbol{x_k}\}_{k \geqslant 0}$ *konvergira ka* $\boldsymbol{x}^*$ *u najviše* n *iteracija.*

Dokaz. *Dokaz većeg dela teoreme dajemo indukcijom. Izrazi pod (b) i (c) trivijalno važe za $k = 0$, dok izraz pod (d) važi za $k = 1$. Pretpostavljamo sada da ovi izrazi važe za neko k (indukcijska hipoteza) i želimo da pokažemo da će važiti i za $k+1$.*

Da bi dokazali (b), pokazujemo prvo da je skup na levoj strani sadržan u skupu na desnoj strani. Iz indukcijske hipoteze, za (b) i (c) imamo respektivno

$$\boldsymbol{r_k} \in span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k \boldsymbol{r_0}\}, \quad \boldsymbol{d_k} \in span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k \boldsymbol{r_0}\},$$

pri tome, množeći drugi izraz sa $\boldsymbol{A}$, dobijamo

$$\boldsymbol{A d_k} \in span\{\boldsymbol{A r_0}, \boldsymbol{A}^2 \boldsymbol{r_0}, ..., \boldsymbol{A}^{k+1} \boldsymbol{r_0}\}. \tag{4.6}$$

Primenjujući $\boldsymbol{r_{k+1}} = \boldsymbol{r_k} + \alpha_k \boldsymbol{A d_k}$, imamo

$$\boldsymbol{r_{k+1}} \in \{\boldsymbol{A r_0}, \boldsymbol{A}^2 \boldsymbol{r_0}, ..., \boldsymbol{A}^{k+1} \boldsymbol{r_0}\}.$$

Kombinujući ovaj izraz sa indukcijskom hipotezom za izraz pod (b), zaključujemo

$$span\{\boldsymbol{r_0}, \boldsymbol{r_1}, ..., \boldsymbol{r_{k+1}}\} \subset span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^{k+1} \boldsymbol{r_0}\}.$$

Da važi i obrnuta inkluzija koristimo indukcijsku pretpostavku za izraz pod (c) da bi izveli

$$\boldsymbol{A}^{k+1} \boldsymbol{r_0} = \boldsymbol{A}(\boldsymbol{A}^k \boldsymbol{r_0}) \in span\{\boldsymbol{A d_0}, \boldsymbol{A d_1}, ..., \boldsymbol{A d_k}\}.$$

Kako znamo da je $\boldsymbol{r_{i+1}} = \boldsymbol{r_i} + \alpha_i \boldsymbol{A d_i}$, *imamo da je* $\boldsymbol{A d_i} = (\boldsymbol{r_{i+1}} - \boldsymbol{r_i})/\alpha_i$, *za* $i = 0, 1, ..., k$, *sledi*

$$\boldsymbol{A}^{k+1}\boldsymbol{r_0} \in span\{\boldsymbol{r_0}, \boldsymbol{r_1}, ..., \boldsymbol{r_{k+1}}\}.$$

Kombinujući ovaj izraz sa indukcijskom pretpostavkom za (b), imamo

$$span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^{k+1}\boldsymbol{r_0}\} \subset span\{\boldsymbol{r_0}, \boldsymbol{r_1}, ..., \boldsymbol{r_{k+1}}\}.$$

Dakle, relacija pod (b) važi i sa zamenom k *sa* $k+1$.

Sada pokazujemo da izraz pod (c) važi sa zamenom k *sa* $k+1$. *Kako znamo da je* $\boldsymbol{d_{k+1}} = -\boldsymbol{r_{k+1}} + \beta_{k+1}\boldsymbol{d_k}$ *trivijalno sledi sledeća jednakost:*

$$span\{\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}, \boldsymbol{d_{k+1}}\} = span\{\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}, \boldsymbol{r_{k+1}}\}.$$

Koristeći indukcijsku hipotezu za izraz pod (c) imamo:

$$span\{\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}, \boldsymbol{r_{k+1}}\} = span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k\boldsymbol{r_0}, \boldsymbol{r_{k+1}}\},$$

što po izrazu pod (b) sledi:

$$span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k\boldsymbol{r_0}, \boldsymbol{r_{k+1}}\} = span\{\boldsymbol{r_0}, \boldsymbol{r_1}, ..., \boldsymbol{r_k}, \boldsymbol{r_{k+1}}\}.$$

Po indukcijskom koraku izraza pod (b) za $k+1$ *imamo:*

$$span\{\boldsymbol{r_0}, \boldsymbol{r_1}, ..., \boldsymbol{r_k}, \boldsymbol{r_{k+1}}\} = span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k\boldsymbol{r_0}, \boldsymbol{A}^{k+1}\boldsymbol{r_0}\}.$$

Sada zaključujemo da je:

$$span\{\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}, \boldsymbol{d_{k+1}}\} = span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^k\boldsymbol{r_0}, \boldsymbol{A}^{k+1}\boldsymbol{r_0}\},$$

i samim tim smo završili dokaz za izraz pod (c).

Sledeće što pokazujemo jeste uslov konjugovanosti pod (d) kada je k *zamenjeno sa* $k+1$. *Množeći* $\boldsymbol{d_{k+1}} = -\boldsymbol{r_{k+1}} + \beta_{k+1}\boldsymbol{d_k}$ *sa* $\boldsymbol{A d_i}, i = 0, 1, ..., k$, *dobijamo*

$$\boldsymbol{d_{k+1}}^T \boldsymbol{A d_i} = -\boldsymbol{r_{k+1}}^T \boldsymbol{A d_i} + \beta_{k+1}\boldsymbol{d_k}^T \boldsymbol{A d_i}. \tag{4.7}$$

Iz definicije β_k, *leva strana izraza (4.7) se anulira za* $i = k$. *Za* $i \leqslant k-1$ *primetimo prvo da indukcijska hipoteza za (d) implicira da su pravci* $\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_k}$ *konjugovani, tako da možemo da primenimo Teoremu 4.1.2. i da zaključimo da je*

$$\boldsymbol{r_{k+1}}^T \boldsymbol{d_i} = 0, \quad za \quad i = 0, 1, ..., k. \tag{4.8}$$

Drugo, ponavljajući primenu izraza pod (c), nalazimo da za $i = 0, 1, ..., k-1$ *sledeća inkluzija važi*

$$\begin{aligned} \boldsymbol{A d_i} \in \boldsymbol{A} span\{\boldsymbol{r_0}, \boldsymbol{A r_0}, ..., \boldsymbol{A}^i\boldsymbol{r_0}\} &= span\{\boldsymbol{A r_0}, \boldsymbol{A}^2\boldsymbol{r_0}, ..., \boldsymbol{A}^{i+1}\boldsymbol{r_0}\} \\ &\subset span\{\boldsymbol{d_0}, \boldsymbol{d_1}, ..., \boldsymbol{d_{i+1}}\}. \end{aligned} \tag{4.9}$$

Kombinujući (4.8) i (4.9) zaključujemo

$$\boldsymbol{r_{k+1}}^T \boldsymbol{A d_i} = 0, \quad za \quad i = 0, 1, ..., k-1,$$

te prvi deo izraza (4.7) sa desne strane se anulira za $i = 0, 1, ..., k-1$. *Zbog indukcijske hipoteze za (d) i drugi deo izraza se takođe anulira sa desne strane izraza (4.7), stoga možemo da zaključimo da je* $\boldsymbol{d_{k+1}^T A d_i} = 0, i = 0, 1, ..., k$. *Dakle, indukcijsko dokazivanje važi i za (d).*

Sledi da je skup pravaca generisan metodom konjugovanih gradijenata zaista skup

konjugovanih pravaca, te Teorema 4.1.1. nam govori da algoritam završava u najviše n iteracija.

Na kraju, dokazujemo izraz pod (a) bez korišćenja indukcije. Kako je skup vektora pravaca konjugovan, imamo da je $\boldsymbol{r_k}^T\boldsymbol{d_i} = 0$*, za sve* $i = 0, 1, ..., k-1$ *i svako* $k = 0, 1, ..., n-1$*. Znamo da je* $\boldsymbol{d_{k+1}} = -\boldsymbol{r_{k+1}} + \beta_{k+1}\boldsymbol{d_k}$*, te je i*

$$\boldsymbol{d_i} = -\boldsymbol{r_i} + \beta_i \boldsymbol{d_{i-1}},$$

pa $\boldsymbol{r_i} \in span\{\boldsymbol{d_i}, \boldsymbol{d_{i-1}}\}$ *za svako* $i = 0, 1, ..., k-1$*. Zaključujemo da je* $\boldsymbol{r_k}^T\boldsymbol{r_i} = 0$ *za svako* $i = 0, 1, ..., k-1$*. Primetimo da je* $\boldsymbol{r_k}^T\boldsymbol{r_0} = -\boldsymbol{r_k}^T\boldsymbol{d_0} = 0$*, po definiciji* $\boldsymbol{d_0}$ *iz algoritma konjugovanih pravaca i* $\boldsymbol{r_k}^T\boldsymbol{d_i} = 0$*, za sve* $i = 0, 1, ..., k-1$*.*

Dokaz ove teoreme se oslanja na činjenicu da je prvi pravac $\boldsymbol{d_0}$ najstrmiji opadajući pravac $-\boldsymbol{r_0}$; u suštini, rezultat ne važi za drugi izbor pravca $\boldsymbol{d_0}$. Kako su gradijenti $\boldsymbol{r_k}$ međusobno ortogonalni, izraz 'metod konjugovanih gradijenata' je pogrešan. To su u stvari pravci pretraživanja, a ne gradijenti, koji su konjugovani u odnosu na matricu $\boldsymbol{A}$.

Možemo da izvedemo nešto praktičniji oblik metoda konjugovanih gradijenata korišćenjem Teoreme 4.1.2 i 4.1.3. Prvo, koristimo $\boldsymbol{d_{k+1}} = -\boldsymbol{r_{k+1}} + \beta_{k+1}\boldsymbol{d_k}$ i $\boldsymbol{r_k}^T\boldsymbol{d_i} = 0, i = 0, 1, ..., k-1$ u formuli $\alpha_k = -\dfrac{\boldsymbol{r_k}^T\boldsymbol{d_k}}{\boldsymbol{d_k}^T\boldsymbol{A}\boldsymbol{d_k}}$, te sada imamo

$$\alpha_k = \frac{\boldsymbol{r_k^T r_k}}{\boldsymbol{d_k^T A d_k}}.$$

Drugo, koristeći činjenicu da je $\boldsymbol{r_{k+1}} = \boldsymbol{r_k} + \alpha_k\boldsymbol{A}\boldsymbol{d_k}$ tj. $\alpha_k\boldsymbol{A}\boldsymbol{d_k} = \boldsymbol{r_{k+1}} - \boldsymbol{r_k}$ i primenjujući još jednom $\boldsymbol{d_{k+1}} = -\boldsymbol{r_{k+1}} + \beta_{k+1}\boldsymbol{d_k}$ i $\boldsymbol{r_k}^T\boldsymbol{d_i} = 0, i = 0, 1, ..., k-1$ pojednostavljujemo formulu za β_{k+1} tako da sada imamo

$$\beta_{k+1} = \frac{\boldsymbol{r_{k+1}^T r_{k+1}}}{\boldsymbol{r_k^T r_k}}.$$

Koristeći ove formule zajedno sa formulom $\boldsymbol{r_{k+1}} = \boldsymbol{r_k} + \alpha_k\boldsymbol{A}\boldsymbol{d_k}$, dolazimo do standardnog metoda konjugovanih gradijenata koji dajemo u nastavku.

Algoritam : Standardni metod konjugovanih gradijenata

Korak 0. Izabrati početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$.

Korak 1. Izračunati $\boldsymbol{r_0} = \boldsymbol{A}\boldsymbol{x_0} - \boldsymbol{b}$. Ukoliko je $\boldsymbol{r_0} = 0$, ili je ispunjen neki drugi kriterijum zaustavljanja, algoritam se zaustavlja i $\boldsymbol{x_0}$ je (približno) rešenje problema. U suprotnom, uzati za $\boldsymbol{d_0} = -\boldsymbol{r_0}$ i ići na Korak 2.

Korak 2. Izrazčunati $\alpha_k = \dfrac{\boldsymbol{r_k^T r_k}}{\boldsymbol{d_k^T A d_k}}$.

Korak 3. Odrediti narednu tačku iterativnog niza $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \alpha_k\boldsymbol{d_k}$.

Korak 4. Izračunati $\boldsymbol{r_{k+1}} = \boldsymbol{r_k} + \alpha_k\boldsymbol{A}\boldsymbol{d_k}$. Ukoliko je $\boldsymbol{r_{k+1}} = 0$ ili je ispunjen neki drugi kriterijum zaustavljanja, algoritam se zaustavlja i tačka $\boldsymbol{x_{k+1}}$ je (približno) rešenje problema. U suprotnom ići na Korak 5.

Korak 5. Izračunati $\beta_{k+1} = \dfrac{\boldsymbol{r_{k+1}^T r_{k+1}}}{\boldsymbol{r_k^T r_k}}$.

Korak 6. Odrediti $(k+1)$-vi konjugovani vektor $\boldsymbol{d_{k+1}} = -\boldsymbol{r_{k+1}} + \beta_{k+1}\boldsymbol{d_k}$. Postaviti $k = k + 1$ i ići na Korak 2.

Primer 4.1.1. *(Rešavanje linearnog sistema primenom standarnog metoda konjugovanih gradijenata). Želimo da rešimo linearni sistem oblika* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$*. Pri tome neka su*

nam poznati matrica $\boldsymbol{A}$ *i vektor* $\boldsymbol{b}$*. Za inicijalnu tačku uzimamo* $(1,1,1,1)^T$*, a za izlazni kriterijum broj iteracija koji dozvoljavamo (tzv. prag tolerancije). Problem rešavamo primenom MATLAB funkcije* `linearni_metod_konjugovanih`*, čiji kod dajemo u nastavku.*

```
function [x]=linearni_metod_konjugovanih(A,b,x0,tol)
% Metod za resavanje linearnog sistema Ax=b
% INPUT
% ======================
% A ....... simetricna pozitivno definitivna matrica funkcije cilja
% b ....... vektor kolona linearnog oblika problema minimizacije
% x0 ...... pocetna tacka metoda
% tol ..... maksimalan broj iteracija koji tolerisemo
% OUTPUT
% ======================
% x ....... matrica vektora svake iteracije (poslednja kolona je optimalno
% resenje problema)
% r ....... matrica reziduala svake iteracije

x=x0;
iter=1;
r=(A*(x(:,1)))-b;
d=-1.*r(:,1);
while (r(:,iter)~=0) & (iter < tol)
    alfa = (r(:,iter)'*r(:,iter))/(d(:,iter)'*A*d(:,iter));
    x = [x,(x(:,iter)+alfa*d(:,iter))];
    r = [r,(r(:,iter)+alfa*A*d(:,iter))];
    beta = (r(:,iter+1)'*r(:,iter+1))/(r(:,iter)'*r(:,iter));
    d = [d,((-1.*r(:,iter+1))+(beta*d(:,iter)))];
    iter=iter+1;
end
display(r);
display(x);
```

Definišemo ulazne parametre i pozivamo datu funkciju narednim komandama:

```
» A=[3 0 1 2;0 4 0 1;1 0 3 0;2 1 0 4];
» b=[2;1;1;1];
» x0=[1;1;1;1];
» linearni_metod_konjugovanih(A,b,x0,30)
```

Izlaz datog koda je:

```
r =
   4.0000 -0.6098 -0.2005 -0.0594 0.0000
   4.0000 0.2439 -0.2344 0.0829 0.0000
   3.0000 0.7805 -0.0265 -0.0688 0.0000
   6.0000 -0.1463 0.3032 0.0188 0.0000

x =
   1.0000 0.3171 0.5513 0.7695 0.7917
   1.0000 0.3171 0.1908 0.3235 0.3056
   1.0000 0.4878 0.1407 0.0539 0.0694
   1.0000 -0.0244 0.0025 -0.2109 -0.2222
```

□

Pored toga što metod konjugovanih gradijenata služi kao alat za rešavanje linearnog sistema, rešava i problem minimizacije strogo konveksne kvadratne funkcije, te u

nastavku dajemo i takav primer. Pri tome, služimo se već postojećim kodom, a jedina modifikacija jeste drugačiji izbor izlaznog kriterijuma.

Primer 4.1.2. *(Primena standardnog metoda konjugovanih gradijenata na rešavanje minimizacije kvadratne funkcije). Posmatramo problem minimizacije funkcije oblika*

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x},$$

gde je matrica $\boldsymbol{A}$ simetrična pozitivno definitna. U našem primeru, neka je vektor $\boldsymbol{b}$ generisan funckijom `randi` *u MATLAB-u dimenzije 7×1, a matrica $\boldsymbol{A}$ simetricna pozitivno definitna matrica dimenzije 7×7 definisana uz pomoćnu matricu B koja je takođe generisana funkcijom* `randi` *u MATLAB-u. Za inicijalnu tačku uzimamo nula vektor (vektor sa svim 0) dimenzije 7×1, a za parametar tolerancije 10^{-6}. Dati problem rešavamo primenom MATLAB funkcije* `cg_kvadratna`*, čiji kod dajemo u nastavku.*

```
function [x]=cg_kvadratna(A,b,x0,tol)
% INPUT
% ======================
% A ....... simetricna pozitivno definitivna matrica funkcije cilja
% b ....... vektor kolona linearnog oblika problema minimizacije
% x0 ...... pocetna tacka metoda
% tol...... prag tolerancije
% OUTPUT
% ========================
% x ....... matrica vektora svake iteracije (poslednja kolona je optimalno
% resenje problema)
% r ....... matrica reziduala svake iteracije

x=x0;
iter=1;
r=(A*(x(:,1)))-b;
d=-1.*r(:,1);
while (norm(r) > tol)
    alfa = (r(:,iter)'*r(:,iter))/(d(:,iter)'*A*d(:,iter));
    x = [x,(x(:,iter)+alfa*d(:,iter))];
    r = [r,(r(:,iter)+alfa*A*d(:,iter))];
    beta = (r(:,iter+1)'*r(:,iter+1))/(r(:,iter)'*r(:,iter));
    d = [d,((-1.*r(:,iter+1))+(beta*d(:,iter)))];
    iter=iter+1;
end
display(r);
display(x);
```

Pre nego što pozovemo MATLAB funckiju `cg_kvadratna`, definišemo matrice $\boldsymbol{A}$,$\boldsymbol{B}$ i vektore $\boldsymbol{b}, \boldsymbol{x_0}$ na sledeći način:

```
» x0 = zeros(7,1);
» b = randi([-10,10],7,1);
» B = randi([-10,10],[8,7]);
» A = B'*B;
```

Dobijeni vektor $\boldsymbol{b}$ je oblika:

```
» b =

   4
   -1
   7
```

```
   6
   -7
   8
   10
```

Dobijena matrica $\boldsymbol{A}$ je dimenzije 7×7 i oblika:

```
» A =
   268 -253 -96 121 -79 79 -41
   -253 415 75   -61 33 -95 97
   -96   75 119   12 52 -38 70
   121   -61 12   245 20 -55 -54
   -79    33 52   20 187   5 -90
   79    -95 -38 -55 5   219 -65
   -41    97 70 -54 -90 -65 243
```

Trivijalno je da je matrica $\boldsymbol{A}$ simetrična, a da je pozitivno definitna lako se može proveriti u softveru MATLAB tako što se funkcijom `eig` ispitaju da li su njeni karakteristični koreni pozitivni.

Pozivamo funkciju `cg_kvadratna` narednom komandom:

```
» cg_kvadratna(A,b,x0,1.e-6)
```

i dobijamo sledeći rezultat

```
r =
   Columns 6 through 8
 0.3884 -0.2745 -0.0000
0.0407 -1.1118 0.0000
0.8949 1.1631 0.0000
0.4949 -1.5180 -0.0000
-0.0296 0.1750 0.0000
-0.8581 0.0820 -0.0000
-0.4089 0.1521 0.0000

x =

   Columns 6 through 8
-0.0448 -0.0714 -0.1629
-0.0141 -0.0305 -0.0615
0.0714 0.0552 -0.0368
0.0726 0.0794 0.1455
-0.0696 -0.0664 -0.0592
0.0837 0.0935 0.1279
0.0297 0.0441 0.0935
```

Primetimo da se metod završio nakon 7 iteracija, što smo mogli i da očekujemo, imajući u vidu da važe uslovi Teoreme 4.1.1. □

4.2 Nelinearni metod konjugovanih gradijenata

Metode konjugovanih gradijenata za kvadratne funkcije mogu biti proširene na opšti slučaj nelinearnih funkcija ako posmatramo funkciju $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$ kao aprok-

simaciju drugog reda dobijenu razvojem polazne, nelinearne funkcije u Tejlorov red. Prisetimo se da je Hesijan kvadratne funkcije $f(\boldsymbol{x})$ konstantan i jednak upravo matrici $\boldsymbol{A}$. U prethodnom delu rada smo videli da ova činjenica značajno olakšava primenu metoda konjugovanih gradijenata. Međutim, u slučaju opšte nelinearne funkcije, Hesijan se mora u svakoj iteraciji ponovo računati, što može biti vrlo zahtevno u računarskom smislu. Ukoliko želimo da konstruišemo efikasnu varijantu metoda konjugovanih gradijenata za minimizaciju nelinearne funkcije u opštem slučaju, poželjno je da izostavimo računanje Hesijana u svakom koraku.

U algoritmu metoda konjugovanih gradijenata za minimizaciju kvadratne funkcije, Hesijan funkcije, tj. matrica $\boldsymbol{A}$ se koristi samo za izračunavanje skalara α_k i β_k. Kako je α_k minimum funkcije $f(\boldsymbol{x_k}+\alpha\boldsymbol{d_k})$, $\alpha \geqslant 0$, za (približno) određivanje α_k možemo koristiti neku od metoda za rešavanje nelinearnih jednačina u jednodimenzionalnom slučaju, kao što su metoda sečice, Njutnova metoda, itd.

Preostaje da rešimo problem računanja koeficijenata β_k bez korišćenja Hesijan matrice $\boldsymbol{A}$. Postoji više načina, što rezultira različitim modifikacijama metoda konjugovanih gradijenata za kvadratnu funkciju. Ove modifikacije za računanje β_k koriste samo vrednosti funkcije i gradijenta u tekućoj tački $\boldsymbol{x_k}$, te mogu biti primenjene za minimizaciju proizvoljne nelinearne funkcije. U ovom poglavlju izložićemo nekoliko modifikacija koje se baziraju na algebarskim transformacijama izraza β_k.

4.2.1 Flečer-Rivsova metoda

Flečer i Rivs su pokazali u svom radu [7] kako proširen metod konjugovanih gradijenata može da se primeni na nelinearne funkcije tako što se naprave dve male promene u algoritmu metoda konjugovanih gradijenata. Kao prva, umesto da koristimo formulu da je $\alpha_k = \dfrac{\boldsymbol{r_k}^T\boldsymbol{r_k}}{\boldsymbol{d_k}^T\boldsymbol{A}\boldsymbol{d_k}}$, za dužinu koraka α_k treba da izvršimo linijsko pretraživanje koje pronalazi približan minimum nelinearne funkcije f duž pravca $\boldsymbol{d_k}$. Druga, rezidual $\boldsymbol{r}$ mora biti zamenjen gradijentnom nelinearne funkcije cilja f. Ove promene dovode do sledećeg algoritma za nelinearnu optimizaciju.

Algoritam : Flečer − Rivsov metod

***Korak* 0.** Izabrati početnu tačku $\boldsymbol{x_0} \in \mathbb{R}^n$ za $k = 0$.

***Korak* 1.** Izračunati $f_0 = f(\boldsymbol{x_0})$, $\nabla f_0 = \nabla f(\boldsymbol{x_0})$. Ukoliko je $\|\nabla f(\boldsymbol{x_0})\| \leqslant \varepsilon$, ili je ispunjen neki drugi kriterijum zaustavljanja, algoritam se zaustavlja i $\boldsymbol{x_0}$ je (približno) rešenje problema. U suprotnom, uzeti za $\boldsymbol{d_0} = -\nabla f(\boldsymbol{x_0})$ i ići na Korak 2.

***Korak* 2.** Izračunati α_k linijskim pretraživanjem funkcije f duž vektora $\boldsymbol{d_k}$.

***Korak* 3.** Odrediti narednu tačku iterativnog niza $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \alpha_k\boldsymbol{d_k}$.

***Korak* 4.** Izračunati $f_{k+1} = f(\boldsymbol{x_{k+1}})$. Ukoliko je $\|\nabla f_{k+1}\| < \varepsilon$ ili je ispunjen neki drugi kriterijum zaustavljanja, algoritam se zaustavlja i tačka $\boldsymbol{x_{k+1}}$ je (približno) rešenje problema. U suprotnom ići na Korak 5.

***Korak* 5.** Izračunati $\beta_{k+1}^{FR} = \dfrac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$.

***Korak* 6.** Odrediti $(k+1)$-vi konjugovani vektor $\boldsymbol{d_{k+1}} = -\nabla f_{k+1} + \beta_{k+1}^{FR}\boldsymbol{d_k}$. Postaviti $k = k + 1$ i ići na Korak 2.

Napomenimo da koristimo notaciju da je $f_k = f(\boldsymbol{x_k})$, kao što smo mogli da primetimo u prethodno definisanom algoritmu i koristićemo dalje u radu.

Ako izaberemo da f bude strogo konveksna, kvadratna funkcija i α_k minimizator dobijen tačnim linijskim pretraživanjem, tada se ovaj algoritam redukuje na linearni metod konjugovanih gradijenata. Flečer-Rivsov metod (ili kraće FR metod) je privla-

čan za velike, nelinearne probleme optimizacije jer svaka iteracija zahteva procenu samo funkcije cilja i njenog gradijenta. Nisu potrebne matrične operacije za računanje koraka, i tek nekoliko vektora je neophodno imati u memoriji.

Kod FR metoda, treba da budemo precizniji u izboru linijskog pretraživanja parametra α_k, jer pravac pretraživanja $\boldsymbol{d_k}$ može i da ne bude opadajući pravac ako α_k ne zadovoljava određene uslove. Diskusiju toga dajemo u nastavku.

Uzimajući unutrašnji proizvod izraza $\boldsymbol{d_{k+1}} = -\nabla f_{k+1} + \beta_{k+1}\boldsymbol{d_k}$ (pri tome $k+1$ zamenjujemo sa k) sa gradijentnim vektorom ∇f_k, dobijamo

$$\nabla f_k^T \boldsymbol{d_k} = -\|\nabla f_k\|^2 + \beta_k \nabla f_k^T \boldsymbol{d_{k-1}}. \tag{4.10}$$

Ako je α_{k-1} lokalni minimizator funkcije f duž pravca $\boldsymbol{d_{k-1}}$, onda je $\nabla f_k^T \boldsymbol{d_{k-1}} = 0$. U tom slučaju, iz (4.10) $\nabla f_k^T \boldsymbol{d_k} < 0$, što implicira da je $\boldsymbol{d_k}$ zaista opadajući pravac. U suprotnom, možemo imati da je $\nabla f_k^T \boldsymbol{d_k} > 0$, implicirajući da je $\boldsymbol{d_k}$ rastući pravac. No, izbegavamo ovu situaciju time što zahtevamo da dužina koraka α_k zadovoljava *jake Volfove uslove*, koje navodimo:

$$f(\boldsymbol{x_k} + \alpha_k \boldsymbol{d_k}) \leqslant f(\boldsymbol{x_k}) + c_1 \alpha_k \nabla f_k^T \boldsymbol{d_k}, \tag{4.11}$$

$$|\nabla f(\boldsymbol{x_k} + \alpha_k \boldsymbol{d_k})^T \boldsymbol{d_k}| \leqslant -c_2 \nabla^T f_k \boldsymbol{d_k}, \tag{4.12}$$

gde $0 < c_1 < c_2 < \dfrac{1}{2}$.

Prvi deo Volfovog uslova, tj. (4.11), poznat je u literaturi kao Armižov uslov.

Primer 4.2.1. *Primenićemo, sada, Flečer-Rivsov metod na poznatu Rozenbrok funkciju. Zato ćemo sada dati MATLAB kod pod nazivom* `FR`*, koji možemo primeniti i na proizvoljnu konveksno kvadratnu funkciju.*

```
function [fmin,xmin,ymin,k] = FR(f,x0,y0)
%Flecer Rivsov metod za resavanje nelinearnog problema - kvadratna
%funkcija
% INPUT
% ======================
% f ....... nelinearna funkcija cilja
% x0 ...... prva koordinata ulaznog vektora
% y0 ...... druga koordinata ulaznog vektora
% OUTPUT
% ======================
% fmin .... optimalna vrednost funkcije
% xmin .... prva koordinata optimalnog resenja
% ymin .... druga koordinata optimalnog resenja

tol=10^-4;  %prag tolerancije
x(1)=x0;y(1)=y0;
gradf=[diff(f,sym('x'));diff(f,sym('y'))]; %gradijent nelinearne funkcije f
d(:,1)=-subs(subs(gradf,'x',x0),'y',y0); %pravac pretrazivanja
k=1;
while and(norm(subs(subs(gradf,'x',x(k)),'y',y(k))) > tol, k < 500)
 rho=0.5;c=0.1; %ulazni parametri za pomocnu funkciju armijo
 alfa=armijo(f,rho,c,x(k),y(k),gradf,d(:,k)); % linijsko pretrazivanje alfe
 x(k+1)=x(k)+alfa*d(1,k);
 y(k+1)=y(k)+alfa*d(2,k);
 beta(k+1)=subs(subs(gradf,'x',x(k+1)),'y',y(k+1))'*subs(subs(gradf,'x',x(k+1)),...
        'y',y(k+1))/((subs(subs(gradf,'x',x(k)),'y',y(k))'*subs(subs(gradf,...
        'x',x(k)),'y',y(k))));
 d(:,k+1)= -subs(subs(gradf,'x',x(k+1)),'y',y(k+1)) + beta(k+1)*d(:,k);
```

```
 k=k+1
end
 k=k
 fmin=subs(subs(f,'x',x(k)),'y',y(k))
 xmin=x(k)
 ymin=y(k)
```

Pomoćnu funkciju koju koristimo u prethodnom kodu, služi za linijsko pretraživanje dužine koraka α koji zadovoljava Armižo uslov. MATLAB kod te funkcije, pod nazivom `armijo`, dajemo u nastavku.

```
function [alfa]=armijo(f,rho,c,xk,yk,gradf,d)
% Linijsko pretrazivanje - Armijo uslov
% INPUT
% ======================
% f ....... nelinearna (kvadratna) funkcija cilja
% xk ...... prva koordinata vektora iz k-te iteracije
% yk ...... druga koordinata vektora iz k-te iteracije
% rho ..... parametar za linijsko pretrazivanje iz intervala (0,1)
% c ....... parametar za linijsko pretrazivanje iz intervala (0,1)
% OUTPUT
% ======================
% alfa .... duzina koraka dobijena linijskim pretrazivanjem

gradf_val=subs(subs(gradf,'x',xk),'y',yk);
alfa=0.5; %inicijalna duzina koraka
fkk=subs(subs(f,'x',xk+alfa*d(1)),'y',yk+alfa*d(2)); %potencijalni minimum
fk=subs(subs(f,'x',xk),'y',yk);
 while and(fkk > fk + c*alfa*(gradf_val)'*d, alfa > 1.0000e-06)
     alfa=rho*alfa;
     fkk=subs(subs(f,'x',xk+alfa*d(1)),'y',yk+alfa*d(2)); %novi potencijalni minimum
 end
end
```

Znamo da je Rozenbrok funkcija oblika:

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2. \tag{4.13}$$

Ako za inicijalni vektor uzmemo $(0.0)^T$ i primenimo FR metod pozivajući MATLAB komandu:

```
» FR(100*(y-x^2)^2+ (1-x)^2,0,0)
```

dobijamo sledeći izlaz koda:

```
» k =
   56
» fmin =
   4.5329e-09
» xmin =
   1.0001
» ymin =
   1.0001
```

Poznato nam je da je optimalno rešenje Rozenbrok funkcije 0, i da se dostiže u tački $(1, 1)^T$. Videli smo da je FR metod nakon 56 iteracija došao do optimalnog rešenja, kao i do optimalne vrednosti funkcije. □

Primer 4.2.2. *Posmatraćemo, sada, funkciju oblika:*

$$f(x,y) = \frac{x^2}{4} + \frac{y^2}{10} - 0.8x - y - 0.3xy - 3. \tag{4.14}$$

Primenjujemo FR metod sa inicijalnom tačkom $(0,0)^T$ tako što pozivamo MATLAB funkciju:

```
» FR(x^2/4+y^2/10-0.8*x-y-0.3*x*y-3,0,0)
```

Dobijeni rezultat je sledeći:

```
» k =
    115
» fmin =
   -58.4000
» xmin =
    45.9966
» ymin =
    73.9945
```

□

4.2.2 Polak-Ribierov metod i njegove modifikacije

Postoje mnoge varijante FR metoda koje se razlikuju jedna od drugih uglavnom po izboru parametara β_k. Bitnu varijantu, koju su predložili Polak i Ribiere, definiše ovaj parametar na sledeći način:

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}. \tag{4.15}$$

Dakle, ako u FR metodu za parametar β_{k+1} koristimo izraz (4.15) dobijamo Polak-Ribierov postupak (kraće PR). Kada je f strogo konveksna, kvadratna funkcija i linijsko pretraživanje je tačno, ove dve metode su identične jer iz činjenice $\boldsymbol{r_k}^T\boldsymbol{r_i} = 0$, za $i = 0, 1, ..., k-1$, gradijenti su međusobno ortogonalni, te je $\beta_{k+1}^{FR} = \beta_{k+1}^{PR}$. Po pitanju primene na opšte nelinearne funkcije sa netačnim linijskim pretraživanjem, ponašanje ovih metoda se značajno razlikuje. Numerički rezultati, koji se pojavljuju u literaturi, nam govore da je algoritam Polav-Ribiere snažniji i efikasniji nego Flečer-Rivsov.

Kod PR algoritma jaki Volfovi uslovi (4.11) i (4.12) ne garantuju da je $\boldsymbol{d_k}$ uvek opadajući pravac. Ako definišemo parametar β kao

$$\beta_{k+1}^{+} = \max\{\beta_{k+1}^{PR}, 0\}, \tag{4.16}$$

dolazimo do algoritma koji nazivamo modifikovani Polak-Ribiere (u oznaci PR+), koji obezbeđuje opadajući pravac.

Postoje mnogi drugi izbori za β_{k+1}, kao na primer Hestenes-Štifel formula, koja definiše parametar na način

$$\beta_{k+1}^{HS} = \frac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^T\boldsymbol{d_k}},$$

što dovodi do algoritma koji je sličan Polak-Ribierovom, kako u pogledu teoretskih svojstava konvergencije, tako i u njihovoj praktičnoj primeni algoritama.

Kasnije ćemo videti da je moguće garantovati globalnu konvergenciju za svaki parametar β_{k+1} koji zadovoljava granicu

$$|\beta_k| \leqslant \beta_k^{FR}, \tag{4.17}$$

za svako $k \geqslant 2$. Ova činjenica sugeriše sledeću modifikaciju metoda PR, koji se dobro pokazao u praksi što možemo videti u literaturama. Za svako $k \geqslant 2$, neka je

$$\beta_k = \begin{cases} -\beta_k^{PR} & \text{ako} \quad \beta_k^{FR} < -\beta_k^{FR} \\ \beta_k^{PR} & \text{ako} \quad |\beta_k^{PR}| \leqslant \beta_k^{FR} \\ \beta_k^{FR} & \text{ako} \quad \beta_k^{PR} > \beta_k^{FR}. \end{cases} \tag{4.18}$$

Algoritam zasnovan na ovoj strategiji označavamo sa FR-PR. Postoje i neke druge predložene varijante metoda konjugovanih gradijenata. Na primer, izbor dva parametara β_{k+1} koja poseduju atraktivna teoretska i računska svojstva jesu:

$$\beta_{k+1} = \frac{\|\nabla f_{k+1}\|^2}{(\nabla f_{k+1} - \nabla f_k)^T \boldsymbol{d_k}} \tag{4.19}$$

i

$$\beta_{k+1} = (\widehat{y}_k - 2\boldsymbol{d_k}\frac{\|\widehat{y}_k\|^2}{\widehat{y}_k^T \boldsymbol{d_k}})^T \frac{\nabla f_{k+1}}{\widehat{y}_k^T \boldsymbol{d_k}}, \quad \widehat{y}_k = \nabla f_{k+1} - \nabla f_k. \tag{4.20}$$

Ova dva izbora parametara β_{k+1} garantuju da je $\boldsymbol{d_k}$ opadajući pravac, pod uslovom da dužina koraka zadovoljava Volfove uslove. Algoritmi konjugovanih gradijenata bazirani na parametrima (4.19) i (4.20) su kompetativni sa Polak-Ribierovim metodom. Kako su oni izvan opsega ovog master rada, nesto više o njima može se naći u literaturi [4] i [9], respektivno.

Primer 4.2.3. *Ideja je da primenimo PR metod na funkcije iste kao i kod FR metoda. No, za to nam je neophodan adekvatan algoritam, implementiran u MATLAB-u pod nazivom* `PR`*, čiji kod dajemo u nastavku.*

```
function [fmin,xmin,ymin,k] = PR(f,x0,y0)
%Polak Ribierov metod za resavanje nelinearnog problema - kvadratna
%funkcija
% INPUT
% ======================
% f ....... nelinearna funkcija cilja
% x0 ...... prva koordinata ulaznog vektora
% y0 ...... druga koordinata ulaznog vektora
% OUTPUT
% ======================
% fmin .... optimalna vrednost funkcije
% xmin .... prva koordinata optimalnog resenja
% ymin .... druga koordinata optimalnog resenja
% k ....... broj iteracija

tol=10^-4; %prag tolerancije
x(1)=x0;y(1)=y0;
gradf=[diff(f,sym('x'));diff(f,sym('y'))]; %gradijent nelinearne funkcije f
d(:,1)=-subs(subs(gradf,'x',x0),'y',y0); %opadajuci pravac u tacki (x0,y0)
k=1;
while and(norm(subs(subs(gradf,'x',x(k)),'y',y(k))) > tol, k < 500)
 rho=0.5;c=0.1; %ulazni parametri za pomocnu funkciju armijo
 alfa=armijo(f,rho,c,x(k),y(k),gradf,d(:,k)); %linijsko pretrazivanje alfe
 x(k+1)=x(k)+alfa*d(1,k);
 y(k+1)=y(k)+alfa*d(2,k);
 beta(k+1)=subs(subs(gradf,'x',x(k+1)),'y',y(k+1))'*(subs(subs(gradf,'x',...
        x(k+1)),'y',y(k+1))-subs(subs(gradf,'x',x(k)),'y',y(k)))/...
        ((subs(subs(gradf,'x',x(k)),'y',y(k))'*subs(subs(gradf,'x',x(k)),...
        'y',y(k))));
```

```
 d(:,k+1)=-subs(subs(gradf,'x',x(k+1)),'y',y(k+1))+beta(k+1)*d(:,k);
 k=k+1
end
 k=k
 fmin=subs(subs(f,'x',x(k)),'y',y(k))
 xmin=x(k)
 ymin=y(k)
```

Funkciju za linijsko pretraživanje dužine koraka α koristimo istu kao i kod FR metoda, te nema potrebe da i ovde navodimo njen kod. Primenjujući PR metod na Rozenbrok funkciju (4.13), sa inicijalnom tačkom $(0,0)^T$, dobijamo sledeći rezultat:

```
» k =
   29
» fmin =
   7.0296e-09
» xmin =
   1.0001
» ymin =
   1.0002
```

Primetimo da se Polak-Ribierov metod daleko bolje pokazao od FR metoda kada je u pitanju Rozenbrok funkcija (pri tome se koristila ista inicijalna tačka) jer se završio nakon 29 iteracija.

Ako primenimo PR metod na funkciju (4.14), sa početnom tačkom $(0,0)^T$, dobijeni izlaz je sledeći:

```
» k =
   500
» fmin =
   -58.3608
» xmin =
   44.7819
» ymin =
   72.0290
```

U ovom slučaju, Flečer-Rivsov metod se pokazao znatno boljim u odnosu na Polak-Ribierov metod iz razloga jer je ispunio traženi uslov nakon 115 iteracija, što je značajno manje od 500 iteracija koliko je bilo neophodno PR metodu.

□

4.2.3 Restart

Implementacija nelinearnih metoda konjugovanih gradijenata uglavnom očuvava njihovu blisku konekciju sa linearnim metodama konjugovanih gradijenata. Obično, kvadratna (ili kubna) interpolacija duž pravca pretraživanja $\boldsymbol{d_k}$ je pripojena u proceduri linijskog pretraživanja. Ta osobina garantuje da kada je f striktno konveksna i kvadratna funkcija, korak α_k je izabran da bude tačan jednodimenzionalni minimizator, te se nelinearni metod konjugovanih gradijenata redukuje na linearni metod.

Naime, ako se radi o tačnom linijskom pretraživanju, metodi konjugovanih pravaca nalaze rešenje u najviše n koraka, ako se primene na kvadratnu funkciju sa pozitivno definitnim Hesijanom kao što smo pokazali u prethodoj sekciji.

Druga modifikacija koja se često koristi kod nelinearnog metoda konjugovanih

gradijenata jeste *restart* iteracije na svakih n koraka postavljajući $\beta_{k+1} = 0$, a to dobijamo postavljanjem vektora pravca na pravac negativnog gradijenta.

Može se pokazati teoretski rezultat vezan za restart koji kaže da restart dovodi do n-koračne kvadratne konvergencije, što znači da je

$$\|\boldsymbol{x_{k+n}} - \boldsymbol{x}\| = \mathcal{O}(\|\boldsymbol{x_k} - \boldsymbol{x}^*\|^2). \tag{4.21}$$

Razmotrimo funkciju f koja je strogo konveksna u okolini rešenje, dok je svuda drugde nekonveksna. Pretpostavimo da algoritam konvergira ka rešenju, te će iteracije na kraju ući u kvadratnu oblast funkcije cilja. U nekom momentu, algoritam će se restartovati u toj oblasti, i od te tačke pa nadalje, njegovo ponašanje će biti ponašanje linearnog metoda konjugovanih gradijenata.

Čak i da funkcija f nije kvadratna u oblasti rešenja, Tejlorova teorema nagoveštava da i dalje može biti približna kvadratnoj, pod uslovom da je neprekidna.

Iako je rezultat (4.21) interesantan sa teorijskog aspekta, ne mora biti i relevantan u praksi, jer nelinearne metode konjugovanih gradijenata mogu biti korisne samo kada su u pitanju rešavanje problema sa velikim n. Restart se možda i ne pojavi kod takvih problema jer se približno rešenje može naći u manje od n koraka. Stoga su nelinearne metode konjugovanih gradijenata nekada implementirane i bez restarta, ili uključuju strategije za restartovanje koje nisu zasnovane na razmatranju o broju iteracija.

Najpoznatija strategija za restart posmatra $\boldsymbol{r_k}^T \boldsymbol{r_i} = 0$, $i = 0, 1, ..., k-1$, što znači da su gradijenti međusobno ortogonalni kada je f kvadratna funkcija. Restart se izvodi svaki put kada su dva uzastopna gradijenta daleko od ortogonalnosti, mereno testom

$$\frac{|\nabla f_k^T \nabla f_{k-1}|}{\|\nabla f_k\|^2} \geqslant \nu, \tag{4.22}$$

pri tome, uobičajna vrednosti za ν je 0.1.

Takođe, formula (4.17) može da se uzme u obzir za strategiju restarta, jer će se $\boldsymbol{d_{k+1}}$ vratiti da bude pravac najstrmijeg pada kad god je β_k^{PR} negativan. Nasuprot (4.22), ovi restarti su daleko ređi jer je β_k^{PR} uglavnom pozitivno.

4.2.4 Ponašanje Flečer-Rivsovog metoda

U ovoj sekciji pažnju posvećujemo algoritmu Flečer-Rivs, dokazujući njegovu globalnu konvergenciju i objašnjavajući neke od njegovih nedostataka.

Naredna lema postavlja uslove na linijsko pretraživanje pod kojim važi da su svi pravci pretraživanja opadajući. Pretpostavlja se da je nivo skup $\mathcal{L} := \{\boldsymbol{x} : f(\boldsymbol{x}) \leqslant f(\boldsymbol{x_0})\}$ ograničen i da je f dva puta diferencijabilna, te tada znamo da postoji dužina koraka α_k koji zadovoljava jake Volfove uslove [13].

Lema 4.2.1. *Pretpostavimo da je Flečer-Rivsov algoritam implementiran sa dužinom koraka α_k koji zadovoljava jak Volfov uslov (4.12) sa $0 < c_2 < \frac{1}{2}$. Tada metod generiše opadajuće pravce $\boldsymbol{d_k}$ koji zadovoljavaju sledeću nejednakost:*

$$-\frac{1}{1-c_2} \leqslant \frac{\nabla f_k^T \boldsymbol{d_k}}{\|\nabla f_k\|^2} \leqslant \frac{2c_2 - 1}{1 - c_2}, \text{ za svako } k = 0, 1, ... \tag{4.23}$$

Dokaz. *Posmatramo funkciju $t(\xi) = (2\xi - 1)/(1 - \xi)$ na intervalu $[0, \frac{1}{2}]$. Primetimo da je $t(0) = -1$ i $t(\frac{1}{2}) = 0$. Da bismo utvrdili da je funkcija t monotono rastuća, tražimo prvi izvod te iste funkcije na intervalu $[0, \frac{1}{2}]$. Vidimo da je $t'(\xi) = \frac{2(1-\xi) + 2\xi - 1}{(1-\xi)^2}$.*

Kada sredimo izraz dobijamo $t'(\xi) = \dfrac{1}{(1-\xi)^2} > 0$, *te zaključujemo da je funkcija* t *monotono rastuća.*
Dakle, kako je $c_2 \in (0, \dfrac{1}{2})$, *imamo*

$$-1 < \frac{2c_2 - 1}{1 - c_2} < 0. \tag{4.24}$$

Samim tim uslov pada $\nabla f_k^T \boldsymbol{d_k} < 0$ *sledi neposredno kada dokažemo (4.23).*

Dokaz dajemo indukcijom. Za $k = 0$, $\frac{\nabla f_k^T \boldsymbol{d_k}}{\|\nabla f_k\|^2}$ *iz (4.23) je -1, te koristeći (4.24) vidimo da su obe nejednakosti u (4.23) zadovoljene. Pretpostavimo sada da (4.23) važi za neko* $k \geqslant 1$. *Iz FR algoritma, koristeći parametre* β_{k+1} *i* $\boldsymbol{d_{k+1}}$, *imamo*

$$\frac{\nabla f_{k+1}^T \boldsymbol{d_{k+1}}}{\|\nabla f_{k+1}\|^2} = -1 + \beta_{k+1} \frac{\nabla f_{k+1}^T \boldsymbol{d_k}}{\|\nabla f_{k+1}\|^2} = -1 + \frac{\nabla f_{k+1}^T \boldsymbol{d_k}}{\|\nabla f_k\|^2}. \tag{4.25}$$

Koristeći uslov linijskog pretraživanja (4.12), imamo

$$|\nabla f_{k+1}^T \boldsymbol{d_k}| \leqslant -c_2 \nabla f_k^T \boldsymbol{d_k},$$

što je ekvivalentno sa

$$c_2 \nabla f_k^T \boldsymbol{d_k} \leqslant \nabla f_{k+1}^T \boldsymbol{d_k} \leqslant -c_2 \nabla f_k^T \boldsymbol{d_k},$$

te u kombinaciji sa (4.25) i pozivajući se na β_{k+1}^{FR}, *dobijamo*

$$-1 + c_2 \frac{\nabla f_k^T \boldsymbol{d_k}}{\|\nabla f_k\|^2} \leqslant \frac{\nabla f_{k+1}^T \boldsymbol{d_{k+1}}}{\|\nabla f_{k+1}\|^2} \leqslant -1 - c_2 \frac{\nabla f_k^T \boldsymbol{d_k}}{\|\nabla f_k\|^2}.$$

Zamenjujući izraz $\nabla f_k^T \boldsymbol{d_k} / \|\nabla f_k\|^2$ *sa levom stranom indukcijske hipoteze (4.23), dobijamo*

$$-\frac{1}{1 - c_2} \leqslant \frac{\nabla f_{k+1}^T \boldsymbol{d_{k+1}}}{\|\nabla f_{k+1}\|^2} \leqslant \frac{2c_2 - 1}{1 - c_2},$$

što implicira da (4.23) takođe važi i za $k + 1$.

Primetimo da se u poslednjem dokazu koristio samo drugi Volfov uslov, dok će prvi biti neophodan kada bude reč o globalnoj konvergenciji. Ograničenje $\nabla f_k^T \boldsymbol{d_k}$ u (4.23) postavlja granicu koliko brzo norma koraka $\|\boldsymbol{d_k}\|$ može da raste i to će ustvari igrati bitnu ulogu u analizi konvergencije.

Naime, Lema 4.2.1 može da se koristi da se objasne i slabosti Flečer-Rivsovog metoda. Raspravljaćemo o tome da ako metod generiše loš pravac i mali korak, onda će najverovatnije sledeći pravac i korak biti loši.

Označimo sa θ_k ugao između $\boldsymbol{d_k}$ i pravca najstrmijeg pada $-\nabla f_k$, definisano sa

$$\cos \theta_k = \frac{-\nabla f_k^T \boldsymbol{d_k}}{\|\nabla f_k\| \|\boldsymbol{d_k}\|}. \tag{4.26}$$

Pretpostavimo da je $\boldsymbol{d_k}$ loš pravac pretraživanja, u smislu da pravi ugao od skoro 90° sa $-\nabla f_k$, što predstavlja da je $\cos \theta_k \approx 0$. Množeći obe strane (4.23) sa $\|\nabla f_k^T\| / \|\boldsymbol{d_k}\|$ i koristeći (4.26), dobijamo

$$\frac{1 - 2c_2}{1 - c_2} \frac{\|\nabla f_k^T\|}{\|\boldsymbol{d_k}\|} \leqslant \cos \theta_k \leqslant \frac{1}{1 - c_2} \frac{\|\nabla f_k^T\|}{\|\boldsymbol{d_k}\|}, \text{ za svako } k = 0, 1, ... \tag{4.27}$$

Iz ovih nejednakosti zaključujemo da je $\cos\theta_k \approx 0$ ako i samo ako

$$\|\nabla f_k\| \ll \|\boldsymbol{d_k}\|.$$

Kako je $\boldsymbol{d_k}$ skoro ortogonalan sa gradijentom, vrlo je verovatno da je korak iz tačke $\boldsymbol{x_k}$ do tačke $\boldsymbol{x_{k+1}}$ jako mali, samim tim $\boldsymbol{x_{k+1}} \approx \boldsymbol{x_k}$. Ako je tako, imamo $\nabla f_{k+1} \approx \nabla f_k$, te po definiciji β_{k+1}^{FR} imamo

$$\beta_{k+1}^{FR} \approx 1.$$

Koristeći aproksimaciju $\nabla f_{k+1} \approx \nabla f_k \ll \|\boldsymbol{d_k}\|$ u definiciji parametra β_{k+1}^{FR}, zaključujemo da je

$$\boldsymbol{d_{k+1}} \approx \boldsymbol{d_k},$$

tako da će se novi pravac pretraživanja malo (ili neće uopšte) poboljšati u odnosu na prethodni. Dakle, ako uslov $\cos\theta_k \approx 0$ važi u nekoj iteraciji k i ako je sledeći korak mali, usledиće dug niz neproduktivnih iteracija.

Polak-Ribierov metod se ponaša sasvim drugačije u ovakvim okruženjima. Ako pravac pretraživanja $\boldsymbol{d_k}$ zadovoljava $\cos\theta_k \approx 0$ za neko k, i ako je naredni korak jako mali, zamenjujući $\nabla f_k = \nabla f_{k+1}$ u (4.15) dobijamo $\beta_{k+1}^{PR} \approx 0$. Samim tim će novi pravac pretraživanja $\boldsymbol{d_{k+1}}$ biti blizu pravca nastrmijeg pada $-\nabla f_{k+1}$ i $\cos\theta_{k+1}$ će biti blizu 1. Dakle, PR algoritam suštinski izvodi restart nakon što naiđe na loš pravac. Isti argument može da se primeni na PR+ algoritam, kao i na HŠ algoritam. Kod FR-PR algoritma, primetimo da je $\beta_{k+1}^{FR} \approx 1$, a $\beta_{k+1}^{PR} \approx 0$, te (4.18) uzima da je $\beta_{k+1} = \beta_{k+1}^{PR}$. Samim tim taj metod izbegava neefikasnosti FR metoda, a ujedno se oslanja na isti za globalnu gonvergenciju.

4.2.5 Globalna konvergencija

Za razliku od linearnog metoda konjugovanih gradijenata, čija svojstva konvergencije su dobro razumljiva i za koji se zna da je optimalan kao što je već bilo reči, nelinearni metodi konjugovanih gradijenata poseduju iznenađujuća svojstva konvergencije. U ovoj sekciji predstavljamo nekoliko glavnih rezultata za Flečer-Rivsov i Polak-Ribierov metod koristeći linijsko pretraživanje.

Za potrebe ovog odeljka, pretpostavljamo sledeće pretpostavke na funkciju cilja.

***Pretpostavke* 4.1**

- Nivo skup $\mathcal{L} := \{\boldsymbol{x} : f(\boldsymbol{x}) \leqslant f(\boldsymbol{x_0})\}$ je ograničen.
- U nekoj otvorenoj okolini $\mathcal{N}$ od $\mathcal{L}$, funkcija cilja f je Lipšic neprekidno diferencijabilna.

Date pretpostavke impliciraju da postoji konstanta $\widehat{\gamma}$ tako da

$$\|\nabla f(\boldsymbol{x})\| \leqslant \widehat{\gamma}, \text{ za svako } \boldsymbol{x} \in \mathcal{L}. \tag{4.28}$$

Glavni matematički alat koji koristimo u ovom odeljku jeste Zoutendijkova teorema [13]. Ona nam govori da pod Pretpostavkama 4.1 iteracija bilo kog linijskog pretraživanja oblika $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \alpha_k \boldsymbol{d_k}$, gde je $\boldsymbol{d_k}$ opadajući pravac, a α_k zadovoljava Volfove uslove (4.11),(4.12) daje konvergenciju reda

$$\sum_{k=0}^{\infty} \cos^2\theta_k \|\nabla f_k\|^2 < \infty. \tag{4.29}$$

Možemo koristiti ovaj rezultat da dokažemo globalnu konvergenciju za algoritme koji se periodično restartuju time što u jednom momentu postavlja da je $\beta_k = 0$. Ako k_1, k_2 i tako dalje, označavaju iteracije u kojima se pojavljuje restart, iz (4.29) imamo

$$\sum_{k=k_1,k_2,\ldots} \|\nabla f_k\|^2 < \infty. \tag{4.30}$$

Ako ne dozvolimo više od $\widehat{n}$ iteracija između restarta, niz $\{k_j\}_{j=1}^{\infty}$ je beskonačan, te iz (4.30) imamo $\lim_{j\to\infty} \|\nabla f_{k_j}\| = 0$. Što znači, podniz gradijenata teži nuli, ili ekvivalentno

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0. \tag{4.31}$$

Teorema 4.2.1. *Pretpostavimo da važe Pretpostavke 4.1 i da je Flečer-Rivsov algoritam implementiran sa linijskim pretraživanjem koji zadovoljava Volfove uslove (4.11) i (4.12) sa $0 < c_1 < c_2 < \frac{1}{2}$. Tada*

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0. \tag{4.32}$$

Dokaz. *Dokaz dajemo kontradikcijom. Pretpostavimo da važi suprotno od (4.32), što implicira da postoji konstanta $\gamma > 0$ tako da*

$$\|\nabla f_k\| \geqslant \gamma, \tag{4.33}$$

za svako dovoljno veliko k. Zamenjujući levom stranom nejednakosti (4.27) u uslovu (4.29), dobijamo

$$\sum_{k=0}^{\infty} \frac{\|\nabla f_k\|^4}{\|\boldsymbol{d_k}\|^2} < \infty. \tag{4.34}$$

Korišćenjem (4.12) i (4.23), dobijamo

$$|\nabla f_k^T \boldsymbol{d_{k-1}}| \leqslant -c_2 \nabla f_{k-1}^T \boldsymbol{d_{k-1}} \leqslant \frac{c_2}{1-c_2} \|\nabla f_{k-1}\|^2. \tag{4.35}$$

Tada, iz definicije β_k^{FR} i definisanosti vektora pretraživanja iz FR algoritma dobijamo

$$\begin{aligned} \|\boldsymbol{d_k}\|^2 &\leqslant \|\nabla f_k\|^2 + 2\beta_k^{FR} |\nabla f_k^T \boldsymbol{d_{k-1}}| + (\beta_k^{FR})^2 \|\boldsymbol{d_{k-1}}\|^2 \\ &\leqslant \|\nabla f_k\|^2 + \frac{2c_2}{1-c_2} \beta_k^{FR} \|\nabla f_{k-1}\|^2 + (\beta_k^{FR})^2 \|\boldsymbol{d_{k-1}}\|^2 \\ &= \big(\frac{1+c_2}{1-c_2}\big) \|\nabla f_k\|^2 + (\beta_k^{FR})^2 \|\boldsymbol{d_{k-1}}\|^2. \end{aligned}$$

Primenjujući ovu relaciju u više navrata i definišući $c_3 = (1+c_2)/(1-c_2) \geqslant 1$, imamo

$$\begin{aligned} \|\boldsymbol{d_k}\|^2 &= c_3 \|\nabla f_k\|^2 + (\beta_k^{FR})^2 (c_3 \|\nabla f_{k-1}\|^2 + (\beta_{k-1}^{FR})^2 (c_3 \|\nabla f_{k-2}\|^2 + \ldots + (\beta_1^{FR})^2 \|\boldsymbol{d_0}\|^2)) \ldots) \\ &= c_3 \|\nabla f_k\|^4 \sum_{j=0}^{k} \|\nabla f_j\|^{-2}, \end{aligned} \tag{4.36}$$

pri tome koristimo činjenicu da je

$$(\beta_k^{FR})^2 (\beta_{k-1}^{FR})^2 \ldots (\beta_{k-i}^{FR})^2 = \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-i-1}\|^4}$$

i $\boldsymbol{d_0} = -\nabla f_0$. Korišćenjem ograničenja (4.28), (4.33) i (4.36) dobijamo

$$\|\boldsymbol{d_k}\|^2 \leqslant \frac{c_3 \widehat{\gamma}^4}{\gamma^2} k,$$

što implicira

$$\sum_{k=1}^{\infty} \frac{1}{\|\boldsymbol{d_k}\|^2} \geqslant \gamma_4 \sum_{k=1}^{\infty} \frac{1}{k}, \tag{4.37}$$

za neku pozitivnu konstantu γ_4*. S druge strane, iz (4.33) i (4.34) imamo*

$$\sum_{k=1}^{\infty} \frac{1}{\|\boldsymbol{d_k}\|^2} < \infty.$$

Ako kombinujemo ovu nejednakost sa (4.37), dobijamo $\sum_{k=1}^{\infty} 1/k < \infty$*, što nije tačno. Stoga, (4.33) ne važi, a tvrdnja (4.32) je dokazana.*

Rezultat globalne konvergencije može biti proširen i na svaki parametar β_k koji zadovoljava (4.17), a posebno za FR-PR metod sa parametrom (4.18).

Naime, može se pokazati da postoje konstante c_4, c_k tako da

$$\cos\theta_k \geqslant c_4 \frac{\|\nabla f_k\|}{\|\boldsymbol{d_k}\|}, \quad \frac{\|\nabla f_k\|}{\|\boldsymbol{d_k}\|} \geqslant c_5 > 0, \quad k = 1, 2, ...$$

te iz (4.29) sledi

$$\lim_{k\to\infty} \|\nabla f_k\| = 0.$$

U suštini, ovaj rezultat može se uspostaviti za Polak-Ribierov metod pod pretpostavkom da je f strogo konveksna funkcija i da se koristi tačno linijsko pretraživanje.

Za generalne (nekonveksne) funkcije, nije moguće dokazati rezultat kao u Teoremi 4.2.1 za Polak-Ribierov algoritam. Sledeća teorema nam govori da PR algoritam može da kruži beskonačno bez pristupanja rešenju, čak i kad se idealno linijsko pretraživanje koristi (pod idealnim mislimo da linijsko pretraživanje vraća vrednost α_k koja je prva pozitivna stacionarna tačka funkcije $t(\alpha) = f(\boldsymbol{x_k} + \alpha \boldsymbol{d_k})$).

Teorema 4.2.2. *Posmatrajmo Polak-Ribierov metod sa parametrom (4.15) i idealnim linijskim pretraživanjem. Tada postoji dva puta neprekidno diferencijabilna funkcija* $f : \mathbb{R}^3 \to \mathbb{R}$ *i početna tačka* $\boldsymbol{x_0} \in \mathbb{R}^3$ *tako da niz gradijenata* $\|\nabla f_k\| >> 0$*.*

Dokaz Teoreme 4.2.2 je prilično kompleksan i prevazilazi opseg ovog rada, te detaljnija analiza, diskusija kao i dokaz može se videti u [16].

5 Zaključak

Ovaj master rad posvećen je gradijentnim metodama i metodama konjugovanih gradijenata, kao dvema bitnim oblastima numeričke analize, alatima u optimizaciji i osnovama za mnogobrojne algoritme. Literatura koja je dostupna je obimna i bilo je nemoguće sve staviti u okvir jednog rada. Kako se koncept ovih metoda tokom vremena razvijao i modifikovao i još uvek se razvija na mnogo načina, glavni cilj rada je bila sistematizacija nekih poznatih rezultata i njihova teorijska i numerička analiza.

Na početku rada definisali smo metode opadajućih pravaca, uz tri moguća izbora dužine koraka: da je konstantan, tačnim linijskim pretraživanjem i linijskim pretraživanjem unazad. Jedan od osnovnih metoda za rešavanje problema optimizacije bez ograničenja je metod najbržeg pada, kojeg smo implementirali kroz primer u softverskom paketu MATLAB i to uzimajući sve tri pomenute dužine koraka. Mogli smo primetiti da rešavanje istog problema primenom metode najbržeg pada sa različitim izborom dužine koraka znatno utiče na konvergencije datog problema. Glavna mana ovog metoda jeste što u svakoj iteraciji treba da se računa prvi izvod posmatrane funkcije cilja $f : \mathbb{R}^n \to \mathbb{R}$, što može biti i prilično kompleksno. Isto tako, pronalaženje izvoda u jednoj iteraciji zahteva neretko veliki broj operacija. U slučaju kada je broj n velik, izračunavanje izvoda postaje komplikovano, a brzina konvergencija opada. Prema tome, kada su funkcije cilja poprilično složene, kao i kod sistema velikih dimenzija, implementacija metode najbržeg pada nije efikasna. Takođe, videli smo na jednom od primera da metod najbržeg pada može da konvergira veoma sporo i kada je uslovni broj velik, pa je jedno od rešenja ovog problema prelazak na skalirani gradijentni metod, koji je opisan i implementiran u ovom radu. Posebnu pažnju smo posvetili i (prigušenom) Gaus-Njutnovom metodu, pomoću kojeg smo rešavali nelinearni problem najmanjih kvadrata, a koji je u osnovi skalirani gradijentni metod. Napokon, čuveni Ferma-Veberov problem smo rešavali Vajsfeldovim metodom, koji takođe pripada klasi gradijentnih metoda.

Metode konjugovanih gradijenata, sa druge strane, prevazilaze problem spore konvergencije koji je karakteristika metode najbržeg pada. Pokazano je da ovaj metod primenjen na sistem linearnih jednačina daje rešenje tog sistema u najviše n koraka, gde je n dimenzija sistema. Nakon toga, u radu je analizirano više metoda za rešavanje problema nelinearne optimizacije. Glavna karakteristika ovih postupaka se ogleda u jednostavnosti i u minimalnoj količini memorije koju zahtevaju. Prvi metod konjugovanih gradijenata za rešavanje problema nelinearne optimizacije bez ograničenja predložili su Flečer i Rivs (FR metod) i u ovom radu smo se bavili i tim metodom i modifikacijom poznatom kao Polak-Ribierov (PR) metod. Mogli smo da vidimo da je PR metod imao bolje performanse u odnosu na FR metod kada je u pitanju njihova primena na Rozenbrok funkciju. U drugom primeru, kada smo PR i FR metod primenili na proizvoljnu, nelinearnu funkciju čiji je minimum veliki negativni broj, a iteracije daleko od minimuma funkcije, tada se FR metod pokazao boljim. Ono što bi moglo da unapredi ove metode jeste takozvana strategija restarta, pogotovo kod PR metoda jer

se dešava da za neke probleme sam metod 'beskonačno kruži' oko rešenja, ne dajući dobru aproksimaciju traženog rešenja. Metod restarta podrazumeva da posle određenog broja iteracija, vektor pravca postavljamo na pravac negativnog gradijenta.

Iako nismo analizirali kompleksnije nelinearne probleme, jasno je da je potreba za modifikacijama pomenutih postupaka konjugovanih gradijenata uvek prisutna i uvek poželjna kako bi dobili algoritme koji imaju manje računarskih operacija, manje skladištenja u memoriji, kraće vreme trajanja algoritma i bolju efikasnost.

Dakle, nema univerzalnog metoda za rešavanje problema optimizacije bez ograničenja, jer sam izbor i ponašanje metoda zavise od posmatrane funkcije cilja. Konstanta je potreba za metodama koji će brzo konvergirati, bez prisustva matrice Hesijana, što je naročito aktuelno za sisteme velikih dimenzija i koji nisu računarski 'skupi', u smislu da su jednostavni za implementaciju i zauzimaju malo računarske memorije. Kako ne postoji idealan postupak, uvek tragamo za onim metodom koji će nam dati zadovoljavajuće kriterijume. Iz tih razloga, gradijentne metode i metode konjugovanih gradijenata se i dalje razvijaju i šire svoju primenu u mnogim oblastima.

Literatura

[1] D. Adnađević, Z. Kadelburg, *Matematička analiza I*, Naučna knjiga, Beograd, 1989.

[2] A. Beck, *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*, Society for Industrial and Applied Mathematics, 2014.

[3] D. P. Bertsekas, W. W. Hager, and O. L. Mangasarian, *Nonlinear programming*, Belmont, MA: Athena Scientific, 1998.

[4] Y.H. Dai, Y. Yuan, *"A nonlinear conjugate gradient method with a strong global convergence property."* SIAM Journal on optimization 10.1 (1999): 177-182.

[5] R. Dimitrijević, *Analiza realnih funkcija više promenljivih*, autor, 1999.

[6] S. Đorđević, *"Izbor parametara kod gradijentnih metoda za probleme optimizacije bez ograničenja."*, Prirodno-matematički fakultet, Departman za matematiku i informatiku, Novi Sad, 2015.

[7] R. Fletcher, C. Reeves, *Function minimization by conjugate gradients*, Comput. J. 7 (1964), pp. 149–154.

[8] Lj. Gajić, *Predavanja iz analize I*, Prirodno-matematički fakultet, Departman za matematiku i informatiku, 2006.

[9] W. W. Hager, H. Zhang. *"A new conjugate gradient method with guaranteed descent and an efficient line search."* SIAM Journal on optimization 16.1 (2005): 170-192.

[10] M.R. Hestenes, E.L. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards 49 (1952), pp. 409– 436.

[11] D. Herceg, N. Krejić, *Numerička analiza*, Prirodno-matematički fakultet, Novi Sad, 1997.

[12] D. G. Luenberger, and Y. Ye, *Linear and nonlinear programming*, Vol. 2. Reading, MA: Addison-wesley, 1984.

[13] J. Nocedal, and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[14] N. Mladenović. *Kontinualni lokacijski problemi*, Matematički institut SANU, 2004.

[15] Z. Pap, *"Projektivni postupci tipa konjugovanih gradijenata za rešavanje nelinearnih monotonih sistema velikih dimenzija."*, Prirodno-matematički fakultet, Departman za matematiku i informatiku, Novi Sad (2019).

[16] M.J.D. Powell, *"Nonconvex minimization calculations and the conjugate gradient method."* Numerical analysis, Springer, Berlin, Heidelberg, 1984. 122-141.

[17] Z. Stanimirović, *Predavanje: Matematičko programiranje i optimizacija*, Matematički fakultet Univerziteta u Beogradu, Beograd

[18] K. Surla, Z. Lozanov-Crvenković, *Operaciona istraživanja*, Prirodno-matematički fakultet, Novi Sad, 2002.

[19] L.N. Trefethen, D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[20] W. L. Winston, J. B. Goldberg, *Operations research: applications and algorithms*, Vol. 3. Belmont eCalif Calif: Thomson/Brooks/Cole, 2004.

Biografija

Katarina Džepina je rođena 27. maja 1994. godine u Novom Sadu. U svom rodnom mestu, Bačkom Jarku, završava osnovnu školu „Slavko Rodić“ 2009. godine. Potom upisuje gimnaziju „Jovan Jovanović Zmaj“ u Novom Sadu, prirodno-matematički smer, koju završava 2013. godine. Studije na Prirodno-matematičkom fakultetu u istom gradu, smer primenjena matematika, modul matematika finansija upisuje odmah posle srednje škole i uspešno ih završava 2017. godine. Iste godine nastavlja master akademske studije na istom fakultetu, takođe smer primenjena matematika, modul matematika finansija. Zaključno sa septembarskim ispitnim rokom 2019. godine polaže sve ispite predviđene planom i programom i time ostvaruje pravo na odbranu master rada. Tokom master studija, imala je priliku da učestvuje na 32. „ECMI Modeling Week“ organizovanog od strane Departmana za matematiku i informatiku u Novom Sadu, 2018. godine. Od maja 2019. godine zaposlena je u IT kući „Synechron“, gde radi na integraciji softvera koji se koristi na finansijskim tržištima.

Novi Sad, 2020 Katarina Džepina

Univerzitet u Novom Sadu
Prirodno–Matematički Fakultet
Ključna Dokumentacijska Informacija

Redni broj:
RBR
Identifikacioni broj:
IBR
Tip dokumentacije: monografska dokumentacija
BF
Tip zapisa: tekstualni štampani materijal
TZ
Vrsta rada: Master rad
VR
Autor: Katarina Džepina
AU
Mentor: dr Goran Radojev
MN
Naslov rada: Gradijentne metode i metode konjugovanih gradijenata
NR
Jezik publikacije: srpski
JP
Jezik izvoda: srpski/engleski
JI
Zemlja publikovanja: Republika Srbija
ZP
Uže geografsko područje: Vojvodina
UGP
Godina: 2020.
GO
Izdavač: autorski reprint
IZ
Mesto i adresa: Novi Sad, Trg Dositeja Obradovića 4
MA
Fizički opis rada: 5 poglavlja, 62 stranica, 19 lit. citata, 2 slike
FO
Naučna oblast: matematika
NO
Naučna disciplina: primenjena matematika
ND
Ključne reči: Gradijentni metod, opadajući pravac, linijsko pretraživanje, metod konjugovanih pravaca
PO
UDK
Čuva se: u biblioteci Departmana za matematiku i informatiku, Prirodno-matematičkog fakulteta, u Novom Sadu
ČU
Važna napomena:
VN
Izvod:
IZ
Datum prihvatanja teme od strane NN veća: 16.10.2020.

DP
Datum odbrane:
DO
Članovi komisije:
KO
Predsednik: dr Zorana Lužanin, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu
Član: dr Goran Radojev, docent, Prirodno-matematički fakultet, Univerzitet u Novom Sadu
Član: dr Sanja Rapajić, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu

University of Novi Sad
Faculty of Sciences
Key Word Documentation

Accession number:
ANO
Identification number:
INO
Document type: monograph type
DT
Type of record: printed text
TR
Contents code: Master thesis
CC
Author: Katarina Džepina
AU
Mentor: Goran Radojev, PhD
MN
Title: Gradient methods and conjugate gradient methods
XI
Language of text: Serbian
LT
Language of abstract: serbian/english
LA
Country of publication: Republic of Serbia
CP
Locality of publication: Vojvodina
LP
Publication year: 2020.
PY
Publisher: author's reprint
PU
Publ. place: Novi Sad, Trg Dositeja Obradovića 4
PP
Physical description: 5 chapters, 62 pages, 19 references, 2 figures
PD
Scientific field: mathematics
SF
Scientific discipline: applied mathematics
SD
Key words: Gradient method, descent direction, line search, conjugate gradient method
UC
Holding data: Department of Mathematics and Informatics' Library, Faculty of Sciences, Novi Sad
HD
Note:
N
Abstract:
AB
Accepted by the Scientific Board on: October 16, 2020
ASB
Defended:

DE
Thesis defend board:
DB
President: Zorana Lužanin, PhD, Full Professor, Faculty of Sciences, University of Novi Sad
Member: Goran Radojev, PhD, Assistant Professor, Faculty of Sciences, University of Novi Sad
Member: Sanja Rapajić, PhD, Full Professor, Faculty of Sciences, University of Novi Sad