Sveučilište u Zagrebu
PMF - matematički odjel

Ante – Toni Vrdoljak

Metode konjugiranih gradijenata

Diplomski rad

Voditelj rada:
prof.dr.sc. N.Antonić

Zagreb, lipanj 2009.

Sadržaj

UVOD

Ljudi optimiziraju. Investitori nastoje kreirati portfelje da izbjegnu prekomjerni rizik dok istovremeno postižu visoke stope povrata. Proizvođači ciljaju na maksimalnu učinkovitost u dizajnu i radu svojih proizvodnih procesa. Inženjeri podešavaju parametre da optimiziraju performanse svojih dizajna.
Priroda optimizira. Fizički sustavi teže stanju minimalne energije. Molekule u izoliranim kemijskim sustavima reagiraju međusobno dokle god se ne smanji potencijalna energija njihovih elektrona. Optimizacija je bitno oruđe i u analizi fizičkih sustava.
Naš cilj u ovom radu će biti predstavljanje najboljih tehnika za rješavanje konačnodimenzionalnih problema optimizacije.
Najviše ćemo se posvetiti metodama i algoritmima konjugiranih gradijenata, što nam je i prvenstveni cilj ovog rada.

U bezuvjetnoj optimizaciji, minimaliziramo ciljnu funkciju koja ovisi o realnim varijablama, bez ograničenja na bilo koju vrijednost ovih varijabli. Matematička formulacija problema je

$$\min_x f(x),$$

gdje je $x \in R^n$ realni vektor sa $n \geq 1$ komponenti i $f : R^n \to R$ glatka funkcija.
Općenito, najbolja situacija je ako uspijemo naći globalni minimum funkcije *f*, mjesto gdje funkcija poprima najmanju vrijednost. Formalna definicija glasi:

x^* je *globalni minimum* ako je $f(x^*) \leq f(x)$ za svaki $x \in R^n$.

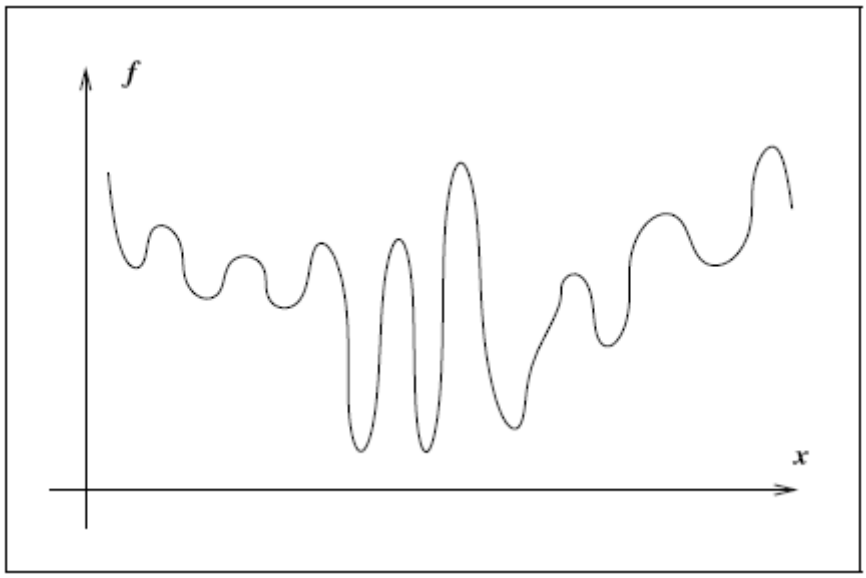
Globalni minimum je najčešće teško pronaći, jer je naše znanje o f obično samo lokalno. Treba imati na umu i da je većina algoritama prikladna za nalaženje samo lokalnih minimuma.

x^* je *lokalni minimum* ako postoji okolina N točke x^* takva da je $f(x^*) \leq f(x)$ za svaki $x \in N$.

Definira se i strogi lokalni minimum:

x^* je *strogi (striktni) lokalni minimum* ako postoji okolina N točke x^* takva da je $f(x^*) < f(x)$ za svaki $x \in N$ i $x \neq x^*$.

Slika (U.1) ilustrira funkciju s mnoštvom lokalnih minimuma.



Slika (U.1) Težak slučaj za globalnu minimizaciju.

Obično je teško naći globalni minimum za ovakve funkcije jer algoritmi mogu biti zadržani oko lokalnih minimuma. U nekim praktičnim primjerima, posebice u kemijskim analizama struktura molekula, potencijal može imati na milijune lokalnih minimuma.

Ponekad imamo i globalno znanje o f koje nam može pomoći pri određivanju lokalnog minimuma. Posebni slučaj su svakako konveksne funkcije, za koje je svaki lokalni minimum ujedno i globalni minimum.

Po gornjim tvrdnjama, čini nam se da je jedini način da otkrijemo da li je x^* lokalni minimum da ispitamo sve točke u njegovoj neposrednoj blizini i uvjerimo se da niti jedna od njih nema manju vrijednost funkcije. No, ako je funkcija f glatka, postoji učinkovitiji i praktičniji način da identificiramo lokalni minimum. Konkretno, ako je f dvaput neprekidno diferencijabilna, možemo reći da je x^* lokalni minimum ispitujući samo gradijent $\nabla f(x^*)$ i Hessian $\nabla^2 f(x^*)$. Matematičko oruđe koje koristimo za proučavanje glatkih funkcija je Taylorov teorem.

TEOREM (U.1) *(Taylorov teorem)*

Pretpostavimo da je $f : R^n \to R$ neprekidno diferencijabilna funkcija i neka je $p \in R^n$. Tada imamo da je

$$f(x+p) = f(x) + \nabla f(x+tp)^T p$$

za neki $t \in (0,1)$. Ako je f dvaput neprekidno diferencijabilna imamo da je

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)\, p \; dt$$

i

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp)\, p, \text{ za neki } t \in (0,1).$$

Nužni uvjeti optimalnosti se izvode tako da pretpostavimo da je x^* lokalni minimum pa zatim provjeravamo činjenice vezane uz $\nabla f(x^*)$ i $\nabla^2 f(x^*)$.

TEOREM (U.2) *(nužni uvjeti prvog reda)*

Ako je x^* lokalni minimum i f neprekidno diferencijabilna funkcija na otvorenoj okolini točke x^*, tada je $\nabla f(x^*) = 0$.

TEOREM (U.3) *(nužni uvjeti drugog reda)*

Ako je x^* lokalni minimum funkcije f i $\nabla^2 f$ postoji i neprekidan je na otvorenoj okolini točke x^*, tada je $\nabla f(x^*) = 0$ i $\nabla^2 f(x^*)$ je pozitivno definitan.

TEOREM (U.4) *(dovoljni uvjeti drugog reda)*

Pretpostavimo da je $\nabla^2 f$ neprekidan na otvorenoj okolini točke x^*, te da je $\nabla f(x^*) = 0$ i $\nabla^2 f(x^*)$ je pozitivno definitan. Tada je x^* strogi lokalni minimum funkcije f.

TEOREM (U.5)

Ako je f konveksna, svaki lokalni minimum x^* je i globalni minimum funkcije f. Ako je f i diferencijabilna, tada je bilo koja stacionarna točka x^* globalni minimum funkcije f.

Ovaj rezultat, baziran na jednostavnom računu, stvara temelje za algoritme optimizacije. Na jedan ili drugi način, svi algoritmi traže točku gdje $\nabla f(\cdot)$ iščezava.

Posljednjih pedesetak godina razvijaju se algoritmi za optimizaciju glatkih funkcija. Doduše, svi imaju i sličnu proceduru: počevši od x_0, algoritmi optimizacije generiraju niz iteracija $\{x_k\}_{k=0}^{\infty}$, a praktički završavaju kada algoritam ili ne može dalje napredovati ili kad se čini da je došao do rješenja s dovoljnom preciznošću. U odlučivanju kako se kretati dalje nakon iteracije x_k, algoritmi koriste informacije o funkciji f u x_k, a po mogućnosti i informacije iz ranijih iteracija $x_0, x_1, \ldots x_{k-1}$. Ove informacije koriste za nalaženje nove iteracije x_{k+1} s nižom vrijednošću funkcije nego li u x_k.

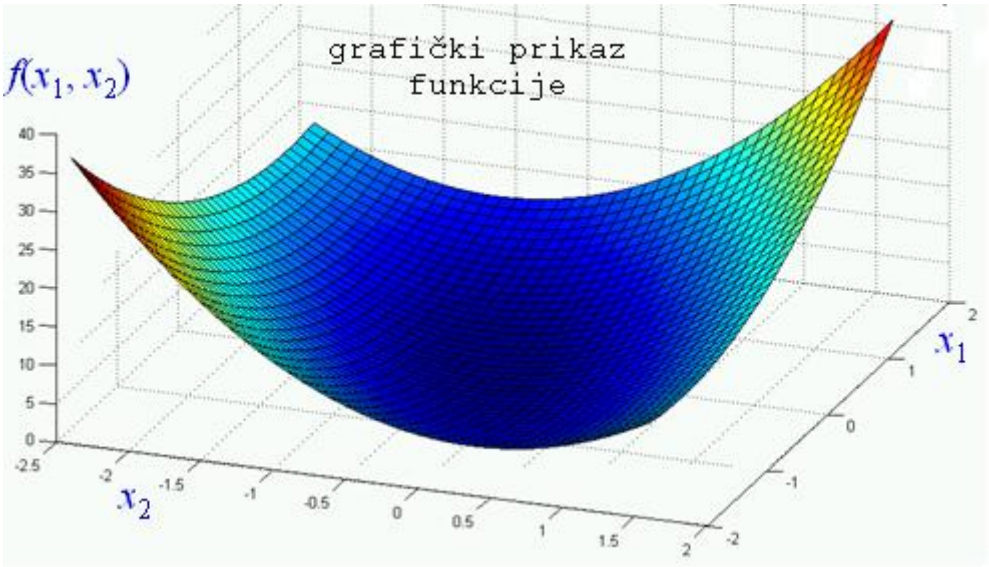
Postoje dvije fundamentalne strategije za pomak iz trenutne pozicije x_k u sljedeću iteraciju x_{k+1}. Većina algoritama koristi jedan od ova dva pristupa: strategiju jednodimenzionalne optimizacije ili strategiju pouzdanog područja.

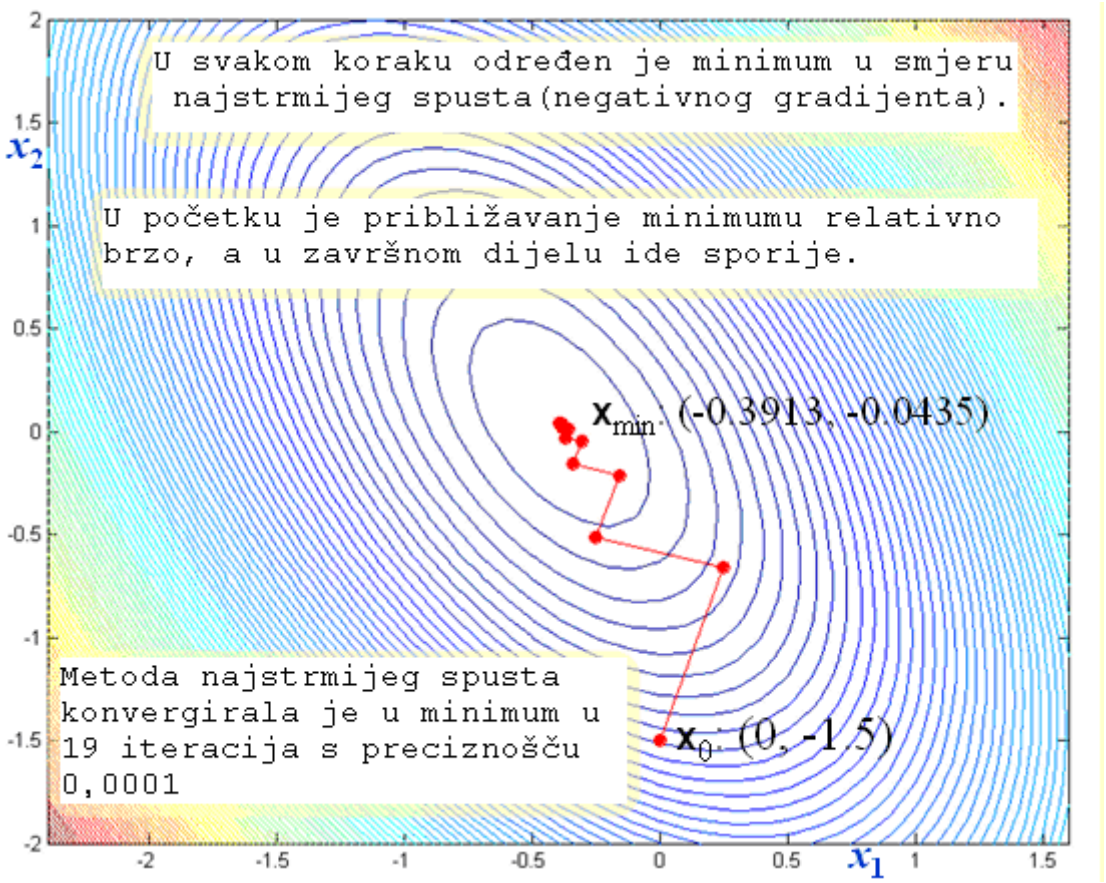
U strategiji jednodimenzionalne optimizacije algoritam odabire smjer p_k i traži duž toga smjera iz trenutne iteracije x_k novu iteraciju s manjom vrijednosti funkcije. Duljina koliko se pomičemo duž p_k može biti aproksimativno pronađena rješavanjem sljedećeg jednodimenzionalnog problema minimizacije:

$$\min_{\alpha>0} f(x_k + \alpha p_k), \qquad \text{(U.1)}$$

gdje je α duljina koraka. Riješimo li (U.1) egzaktno, dobit ćemo maksimalnu korist iz smjera p_k, ali egzaktna minimizacija može biti skupa i često je nepotrebna. Umjesto toga, algoritam tražene linije generira ograničen broj pokusnih duljina koraka dok ne nađe jedan koji aproksimira minimum (U.1). Tada se na novom mjestu računaju nova tražena linija i duljina koraka, i proces se ponavlja.
Pogledajmo ponašanje metode najbržeg silaska na primjeru sljedeće kvadratne funkcije dviju varijabli:

$$f(x_1, x_2) = 1 + x_1 + 3x_2 + 2x_1^2 + 3x_1x_2 + 4x_2^2.$$





U drugoj strategiji, poznatoj kao strategija pouzdanog područja, informacije skupljene o *f* se koriste za konstrukciju probne funkcije m_k čije je ponašanje u blizini trenutne pozicije x_k slično kao i kod aktualne ciljne funkcije *f*. Kako probna funkcija m_k ne mora biti dobra aproksimacija funkcije f kada je *x* daleko od x_k, restringiramo potragu za minimumom probne funkcije m_k na neko područje oko x_k. Drugim riječima, nalazimo korak *p*, tako što aproksimativno riješimo sljedeći potproblem:

$$\min_p m_k(x_k + p), \quad \text{gdje } x_k + p \text{ leži unutar pouzdanog područja.} \qquad \text{(U.2)}$$

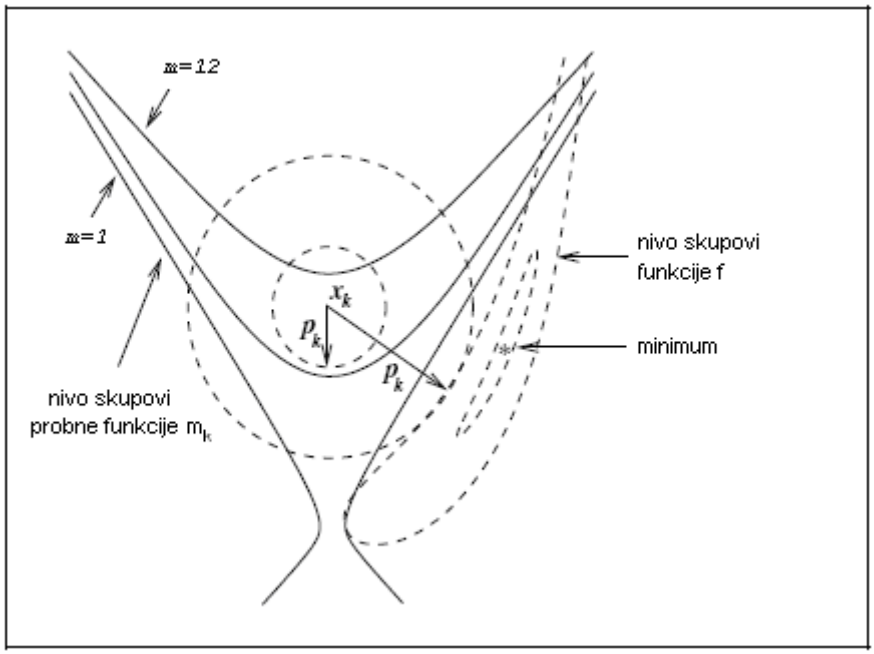
Ako moguće rješenje ne proizvodi dovoljan pad funkcije f, zaključujemo da je pouzdano područje preveliko, pa ga smanjujemo i ponovno rješavamo gornju jednadžbu.
Probna funkcija m_k iz (U.2) je obično definirana kao kvadratna funkcija oblika

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

gdje su $f_k, \nabla f_k$ *i* B_k redom skalar,vektor i matrica. Ovakav zapis povlači da su f_k *i* ∇f_k izabrani da budu vrijednost funkcije i vrijednost gradijenta u točki p_k. Matrica B_k je, ili Hessian $\nabla^2 f_k$, ili neka aproksimacija istog.

Uzmimo da je zadana ciljna funkcija $f(x) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2$. U točki $x_k = (0,1)$ gradijent i Hessian su redom

$$\nabla f_k = \begin{bmatrix} -2 \\ 20 \end{bmatrix}, \qquad \nabla^2 f_k = \begin{bmatrix} -38 & 0 \\ 0 & 20 \end{bmatrix}.$$



Slika (U.2) Dva moguća pouzdana područja (krugovi) i njihovi odgovarajući koraci p_k. Pune linije su nivo skupovi probne funkcije m_k.

Na *slici(U.2)* se vide nivo skupovi ciljne funkcije f i naznačeni su nivo skupovi probne funkcije m_k s vrijednostima od 1 do 12. Primijetimo da svaki put kad smanjimo vrijednost pouzdanog područja, korak od x_k do novog rješenja je kraći i obično pokazuje u drugačijem smjeru od prethodnog rješenja.
Metoda jednodimenzionalne optimizacije i pouzdanog područja se razlikuju u poretku po kojem odabiru smjer i udaljenost do nove iteracije. Metoda jednodimenzionalne optimizacije počinje tako da fiksiramo p_k i odredimo odgovarajuću udaljenost, duljinu koraka α_k. U metodi pouzdanog područja, prvo odabiremo maksimalnu udaljenost – radijus pouzdanog područja Δ_k – te onda tražimo smjer i korak koji će napraviti najbolje poboljšanje na ovoj udaljenosti. Ako se ovaj korak pokaže nedovoljno dobar, smanjimo Δ_k i pokušamo ponovno. Sada ćemo nešto reći i o smjerovima silaska za metode jednodimenzionalne minimizacije.

Smjer najbržeg silaska $-\nabla f_k$ je najočitiji izbor za pretražni smjer u metodi tražene linije. Jasno je da između svih smjerova po kojima smo se mogli pomaknuti od x_k, ovo je onaj duž

kojeg f najbrže silazi. Da bismo provjerili ovu izjavu, pozvat ćemo se na Taylorov teorem koji nam kaže da za bilo koji smjer silaska i parametar duljine koraka α imamo

$$f(x_k + \alpha p) = f(x_k) + \alpha p^T \nabla f_k + \frac{1}{2}\alpha^2 p^T \nabla^2 f(x_k + tp)\, p, \quad \text{za neki } t \in (0, \alpha).$$

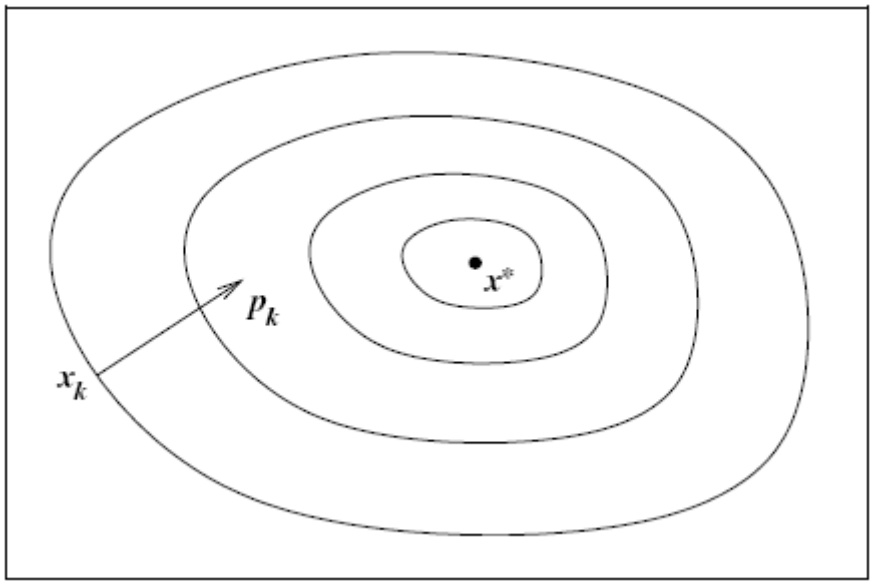
Stopa promjene funkcije f duž smjera p u x_k je jednostavno koeficijent α, konkretno $p^T \nabla f_k$. Jedinični smjer *p* najbržeg silaska je rješenje problema

$$\min_{\|p\|=1} p^T \nabla f_k .$$

Kako je $p^T \nabla f_k = \|p\| \|\nabla f_k\| \cos\theta$, gdje je θ kut između *p i* ∇f_k, lako se vidi da je minimum dosegnut kada je $\cos\theta = -1$ i

$$p = -\nabla f_k / \|\nabla f_k\| .$$

Ovaj smjer je normalan na nivo skupove funkcije, što i vidimo iz *slike (U.3)* .



Slika (U.3) Smjer najbržeg silaska za funkciju dviju varijabli.

Sljedeći važni smjer silaska, možda čak i najvažniji od svih, je Newtonov smjer. Ovaj smjer je izveden iz Taylorovih aproksimacija za $f(x_k + p)$:

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k\, p \overset{def}{=} m_k(p)$$

Ako pretpostavimo da je $\nabla^2 f_k$ pozitivno definitan, dobivamo da je Newtonov smjer traženi vektor *p* koji minimizira $m_k(p)$. Ako stavimo da je gradijent $m_k(p)$ nula, dobivamo sljedeću eksplicitnu formulu:

$$p_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k .$$

Newtonov smjer je pouzdan kada razlika između prave funkcije $f(x_k + p)$ i njezine kvadratne probne funkcije $m_k(p)$ nije prevelika.
Newtonov smjer može biti korišten u metodi jednodimenzionalne optimizacije kada je $\nabla^2 f_k$ pozitivno definitan, pa u tom slučaju imamo

$$\nabla f_k^T p_k^N = -p_k^{N\,T} \nabla^2 f_k p_k^N \le -\sigma_k \left\| p_k^N \right\|^2 ,$$

za neki σ_k >0. Ukoliko gradijent ∇f_k (pa stoga i korak p_k^N) nije nula, imamo da je $\nabla f_k^T p_k^N < 0$, pa je Newtonov smjer upravo smjer silaska.
Za razliku od smjera najbržeg silaska, s Newtonovim smjerom se povezuje 'prirodna' duljina koraka 1. Većina metoda koja koristi Newtonov smjer ima implementiranu duljinu koraka 1 i prilagođavaju ga jedino ako ne daje zadovoljavajuće smanjenje vrijednosti funkcije *f*.
Metode koje koriste Newtonov smjer brzo lokalno konvergiraju. Čim je okolina rješenja dosegnuta, konvergencija visoke točnosti se postiže u samo nekoliko iteracija. Ono što najviše opterećuje Newtonovu metodu je potreba za računanjem Hessiana $\nabla^2 f_k$, što je skup proces podložan greškama.

Kvazi-Newtonova metoda smjerova silaska omogućava nam zanimljivu alternativu Newtonovoj metodi, jer ne zahtijeva računanje Hessiana, a ipak ima visoku brzinu konvergencije. Umjesto pravog Hessiana $\nabla^2 f_k$ koristimo aproksimaciju B_k, koja se ažurira nakon svakog koraka sa skupljenim znanjem iz prošlog koraka. Nije teško pokazati, uz pomoć Taylorova teorema, da se kvazi-Newtonov smjer dobiva upotrebom B_k umjesto egzaktnog Hessiana i to sljedećom relacijom:

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

Neke praktične implementacije kvazi-Newtonove metode izbjegavaju potrebu da faktoriziraju B_k kod svake iteracije tako da ažuriraju inverz matrice B_k, umjesto same B_k.

Posljednja klasa smjerova silaska je generirana metodama nelinearnih konjugiranih gradijenata. Imaju sljedeći oblik

$$p_k = -\nabla f(x_k) + \beta_k p_{k-1} ,$$

gdje je β_k skalar koji osigurava da su p_k i p_{k-1} konjugirani; o tome ćemo raspravljati u poglavljima 1.1 i 1.2. Metoda konjugiranih gradijenata je orginalno zamišljena za rješavanje sustava linearnih jednadžbi $Ax = b$, gdje je *A* kvadratna, simetrična, pozitivno definitna matrica, *x* nepoznati vektor i *b* poznati vektor. Ovaj zadatak možemo ekvivalentno prikazati i kao problem minimizacije konveksnih kvadratnih funkcija:

$$\min \phi(x) \overset{def}{=} \frac{1}{2} x^T Ax - b^T x .$$

Općenito, nelinearne metode konjugiranih smjerova su puno efikasnije od metode najbržeg silaska i često su jednostavne za računanje. S druge strane, ne postižu brzu konvergenciju kao Newtonova i kvazi-Newtonova metoda, ali imaju tu prednost da ne zahtijevaju spremanje matrica. O tome ćemo detaljnije diskutirati u poglavljima 1.1 i 1.2.

Podsjetimo se i sljedećih definicija.

Matrica *A* je pozitivno definitna, ako za svaki ne-nul vektor *x* vrijedi da je

$$x^T Ax > 0 .$$

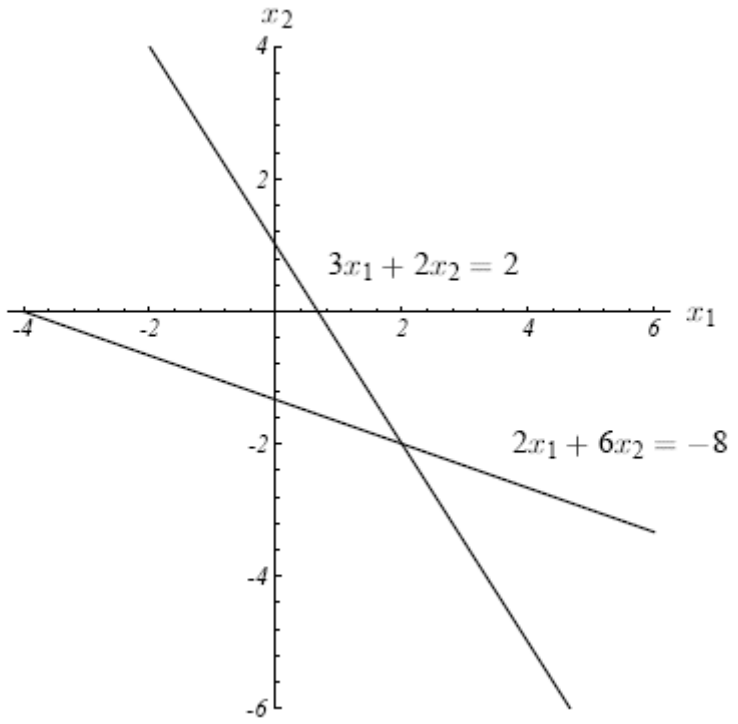
Kvadratna forma je sljedeća funkcija

$$f(x) = \frac{1}{2} x^T Ax - b^T x + c ,$$

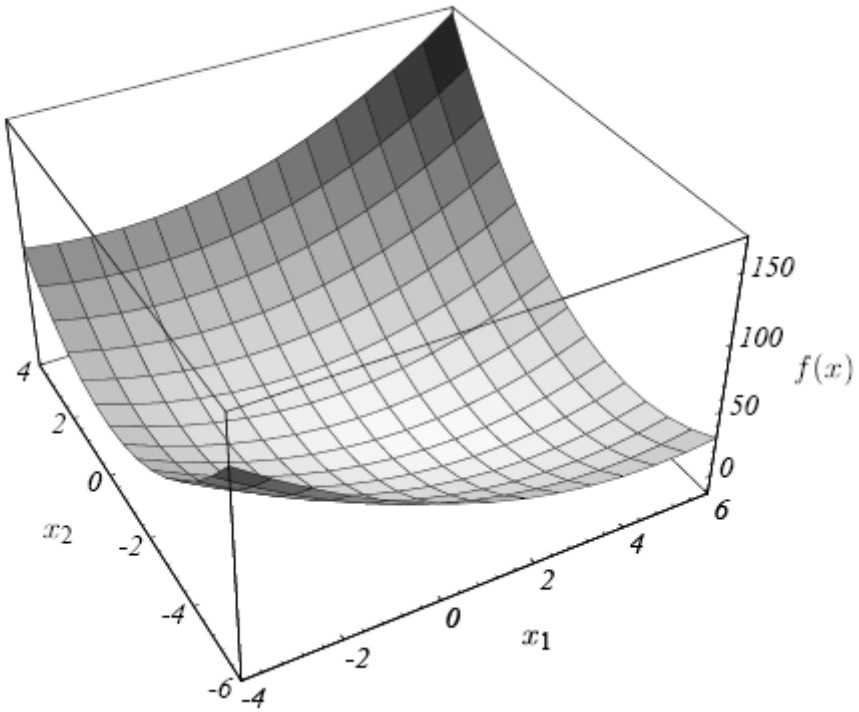
gdje je *A* matrica, *x* i *b* vektori, a *c* skalarna konstanta. Pokazat ćemo da, ako je *A* simetrična, pozitivno definitna, onda je *f(x)* minizirana rješenjem sustava $Ax = b$.
Ideje ćemo demonstrirati na jednostavnom primjeru gdje je

$$A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, \qquad b = \begin{bmatrix} 2 \\ -8 \end{bmatrix}, \qquad \text{c=0.}$$

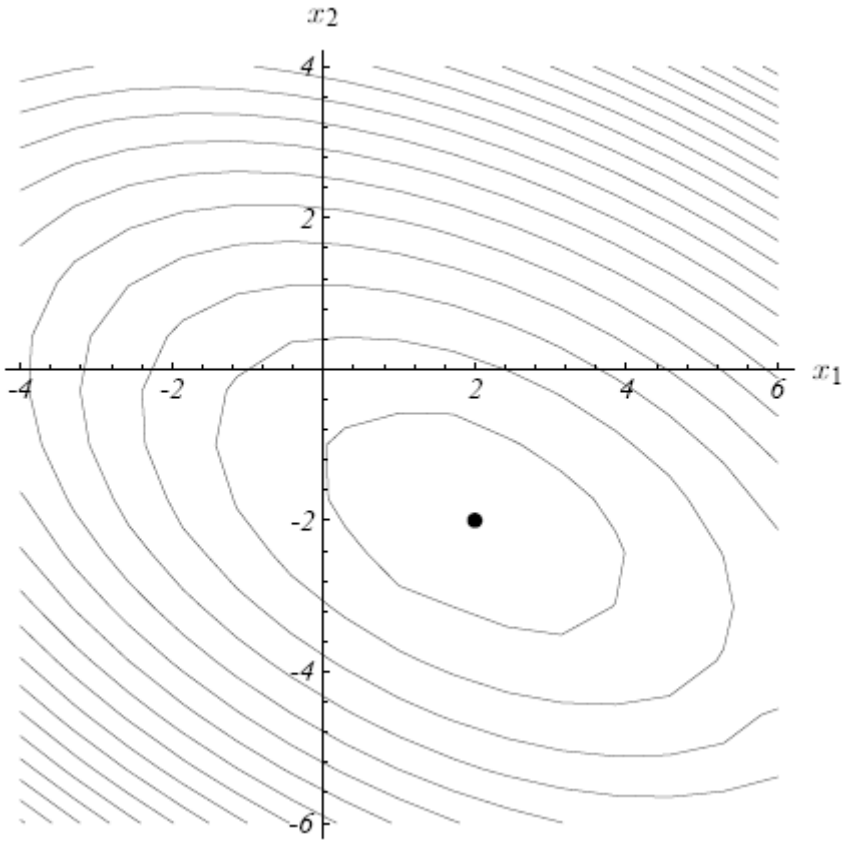
Sustav $Ax = b$ je ilustiran *slikom (U.4).* Općenito, rješenje *x* leži na presjeku *n* hiperravnina. Za ovaj problem rješenje je $x = [2,-2]^T$. Odgovarajuća kvadratna forma se pojavljuje na *slici (U.5)*. Nivo skupovi iste funkcije su prikazani na *slici (U.6).*



Slika (U.4) Primjer dvodimenzionalnog linearnog sustava. Rješenje se nalazi na presjeku ovih pravaca.



Slika (U.5) Graf kvadratne forme *f(x)*. Minimum funkcije *f(x)* je rješenje sustava $Ax = b$.



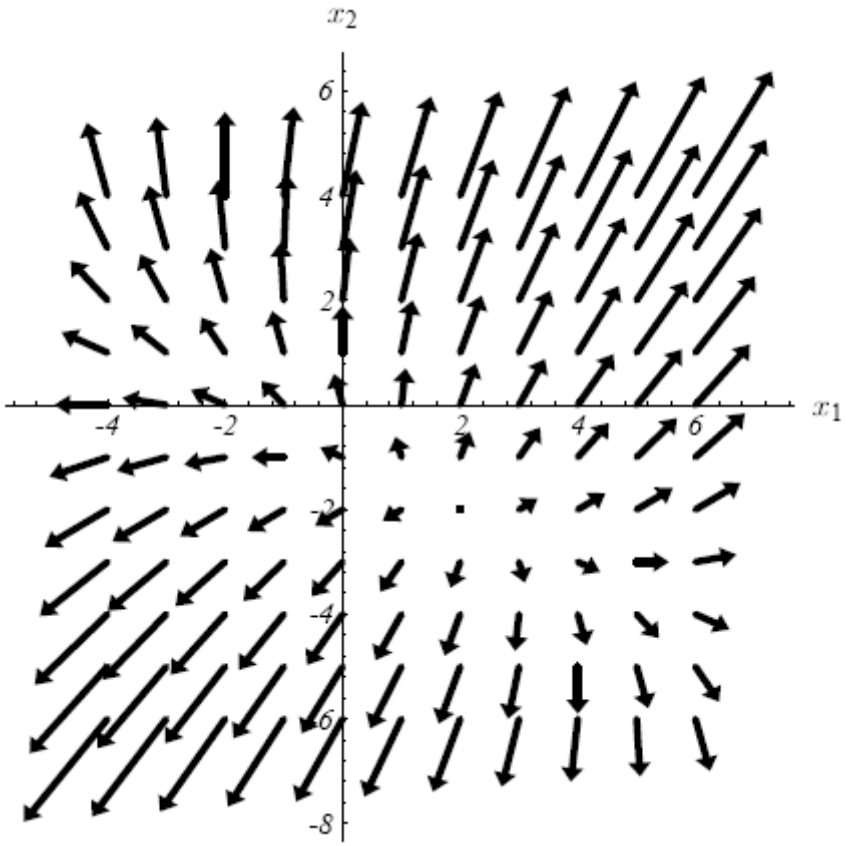
Slika (U.6) Nivo skupovi kvadratne forme. Svaka elipsoidna krivulja ima konstantne vrijednosti *f(x)*.

A je pozitivno definitna pa je površina definirana s *f(x)* oblikovana kao paraboloidna zdjela. Gradijent kvadratne forme je definiran kao

$$f'(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x2} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix}$$

Gradijent je polje vektora koje, za dani *x*, pogađa smjer najvećeg rasta *f(x)*.

Slika (U.7) ilustrira vektore gradijenta za $f(x)=\frac{1}{2}x^T Ax - b^T x + c$ i dane konstante od gore. Na dnu paraboloidne zdjele gradijent je nula. Dakle, možemo minimizirati *f(x)* ako postavimo da je $f'(x)$ jednak nuli.



Slika (U.7) Gradijent $f'(x)$ kvadratne forme. Za svaki *x*, gradijent gleda u smjeru najbržeg silaska od *f(x)* i ortogonalan je nivo skupovima kvadratne forme.

Hessian kvadratne forme je definiran kao

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

Rezimirajmo:

Algoritme za nelinearnu optimizaciju svrstavamo u:

a) one koji zahtijevaju samo poznavanje vrijednosti funkcije u svakoj točki,
b) one koji osim vrijednosti funkcije u svakoj točki zahtijevaju i poznavanje vektora gradijenta,
c) one koje zahtijevaju i poznavanje Hessiana.

Postupci koji zahtijevaju samo poznavanje vrijednosti funkcije u svakoj točki:

- u najjednostavnijem postupku naizmjence se optimizira po jedna varijabla, dok se vrijednosti ostalih ne mijenjaju. Minimum po jednoj varijabli određuje se pretraživanjem u odabranim točkama i računanjem minimuma funkcije koja kroz njih prolazi. Takvi postupci su općenito spori i dolaze u obzir samo kad je računanje gradijenta prezahtjevno.

Postupci koji osim vrijednosti funkcije u svakoj točki zahtijevaju i poznavanje vektora gradijenta:

- metoda najbržeg silaska,
- metode konjugiranih gradijenata.

Postupci koji zahtjevaju i poznavanje Hessiana:

- Newtonova metoda,
- kvazi-Newtonova metoda.

METODE KONJUGIRANIH GRADIJENATA

Naše zanimanje za metode konjugiranih gradijenata je dvostuko. Prvenstveno, one su među najkorisnijim tehnikama za rješavanje velikih sustava linearnih jednadžbi. Kao drugo, mogu se prilagoditi za rješavanje nelinearnih problema optimizacije. Bitne značajke obiju, linearnih i nelinearnih, metoda konjugiranih gradijenata je ono čime ćemo se baviti u ovom radu.

Linearna metoda konjugiranih gradijenata je prestavljena prvi put 50-tih godina prošlog stoljeća (Hestenes i Stiefel) kao iterativna metoda za rješavanje linearnih sustava pozitivno definitnih matrica. Provođenje linearne metode konjugiranih gradijenata je određeno raspodjelom svojstvenih vrijednosti koeficijenata matrice. Kod transformiranja, ili prekondicioniranja, linearnog sustava, možemo napraviti ovu raspodjelu pogodnijom i značajno unaprijediti konvergenciju metode. Prekondicioniranje ima presudnu ulogu u poboljšanju praktičnih izvedbi metoda konjugiranih gradijenata. Naše proučavanje linearne metode konjugiranih gradijenata će istaknuti ona svojstva ove metode koja su bitna u optimizaciji.

Prvu nelinearnu metodu konjugiranih gradijenata su uveli Fletcher i Reeves nekoliko godina poslije Hestenesa i Stiefela. To je jedna od najranijih poznatih tehnika rješavanja velikih nelinearnih problema optimizacije. Tijekom godina razvile su se mnoge varijante ove metode su se razvile, a neke su i široko korištene u praksi. Glavne karakteristike ovih algoritama su što ne zahtijevaju spremanje matrice i brže su od metode najbržeg silaska.

1.1 LINEARNA METODA KONJUGIRANIH GRADIJENATA

U ovom dijelu ćemo opisati linearnu metodu konjugiranih gradijenata i diskutirati njena osnovna svojstva konvergencije. Radi jednostavnosti, ispustiti ćemo pridjev 'linearna'. Metoda konjugiranih gradijenata je iterativna metoda za rješavanje linearnog sustava jednadžbi

$$Ax = b \tag{1.1}$$

gdje je A simetrična, pozitivno definitna matrica dimenzije $n \times n$. Problem (1.1) možemo prikazati ekivalentno kao problem minimizacije

$$\min \phi(x) \overset{def}{=} \frac{1}{2} x^T Ax - b^T x \ , \tag{1.2}$$

gdje obje jednadžbe, (1.1) i (1.2) , naravno imaju, isto jedinstveno rješenje. Ekvivalencija će nam dopustiti da interpretiramo metodu konjugiranih gradijenata, bilo kao algoritam za rješavanje sustava linearnih jednadžbi, bilo kao minimizaciju konveksnih kvadratnih funkcija. Zapišimo da je gradijent funkcije ϕ jednak pogreški pri rješavanju sustava linearnih jednadžbi koju ćemo ubuduće zvati rezidualom

$$\nabla \phi(x) = Ax - b \overset{def}{=} r(x) \tag{1.3}$$

Posebno, za $x = x_k$ imamo da je

$$r_k = Ax_k - b \, . \tag{1.4}$$

Metoda konjugiranih smjerova

Jedno od bitnih svojstava metode konjugiranih gradijenta je mogućnost da generira skup vektora sa svojstvom poznatim kao konjugacija. Skup ne-nul vektora $\{p_0, p_1, ..., p_l\}$ je konjugiran s obzirom na simetričnu pozitivno definitnu matricu *A* ako je

$$p_i^T A p_j = 0, \qquad \text{za svaki } i \neq j \, . \tag{1.5}$$

Bilo koji skup vektora koji zadovoljava ovo svojstvo je nužno i linearno nezavisan. Važnost konjugacije je u činjenici da možemo minimizirati $\phi(\cdot)$ u *n* koraka tako da ga sukcesivno minimiziramo duž individualnih smjerova u konjugiranom skupu. Za provjeru ove tvrdnje promatramo sljedeću metodu konjugiranih smjerova. (Razlika između metode konjugiranih gradijenata i metode konjugiranih smjerova će postati jasna u postupku dalje.) Neka je dana početna točka $x_0 \in R^n$ i skup konjugiranih smjerova $\{p_0, p_1, ..., p_{n-1}\}$. Generiramo niz $\{x_k\}$ tako da stavimo

$$x_{k+1} = x_k + \alpha_k p_k , \tag{1.6}$$

gdje je α_k jednodimenzionalan minimum kvadratne funkcije $\phi(\cdot)$ duž $x_k + \alpha p_k$ dan eksplicitno sa

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} . \tag{1.7}$$

Dobili smo sljedeći rezultat:

TEOREM 1.1

Za svaki $x_0 \in R^n$, konačan niz iteracija $\{x_k\}$ generiran algoritmom konjugiranih smjerova (1.6) i (1.7), konvergira rješenju *x** linearnog sustava jednadžbi (1.1) u najviše *n* koraka za *k = 0,1,...n.*

Dokaz: Kako su smjerovi $\{p_i\}$ linearno nezavisni oni moraju razapinjati cijeli prostor R^n. Stoga, razliku između x_0 i rješenja *x** možemo zapisati na sljedeći način :

$$x^* - x_0 = \sigma_0 p_0 + \sigma_1 p_1 + ... + \sigma_{n-1} p_{n-1},$$

za neki izbor skalara σ_k. Pomnožimo li ovaj izraz sa $p_k^T A$ i koristeći svojstvo konjugacije (1.5) dobivamo

$$\sigma_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k} . \tag{1.8}$$

Sada još trebamo pokazati da se koeficijenti σ_k podudaraju sa duljinom koraka α_k generiranih formulom (1.7).
Ako je x_k generiran algoritmima (1.6) i (1.7) tada imamo da je

$$x_k = x_0 + \alpha_0 p_0 + \alpha_1 p_1 + ... + \alpha_{k-1} p_{k-1} .$$

Množeći ovaj izraz s $p_k^T A$ i koristeći svojstvo konjugacije dobivamo da je

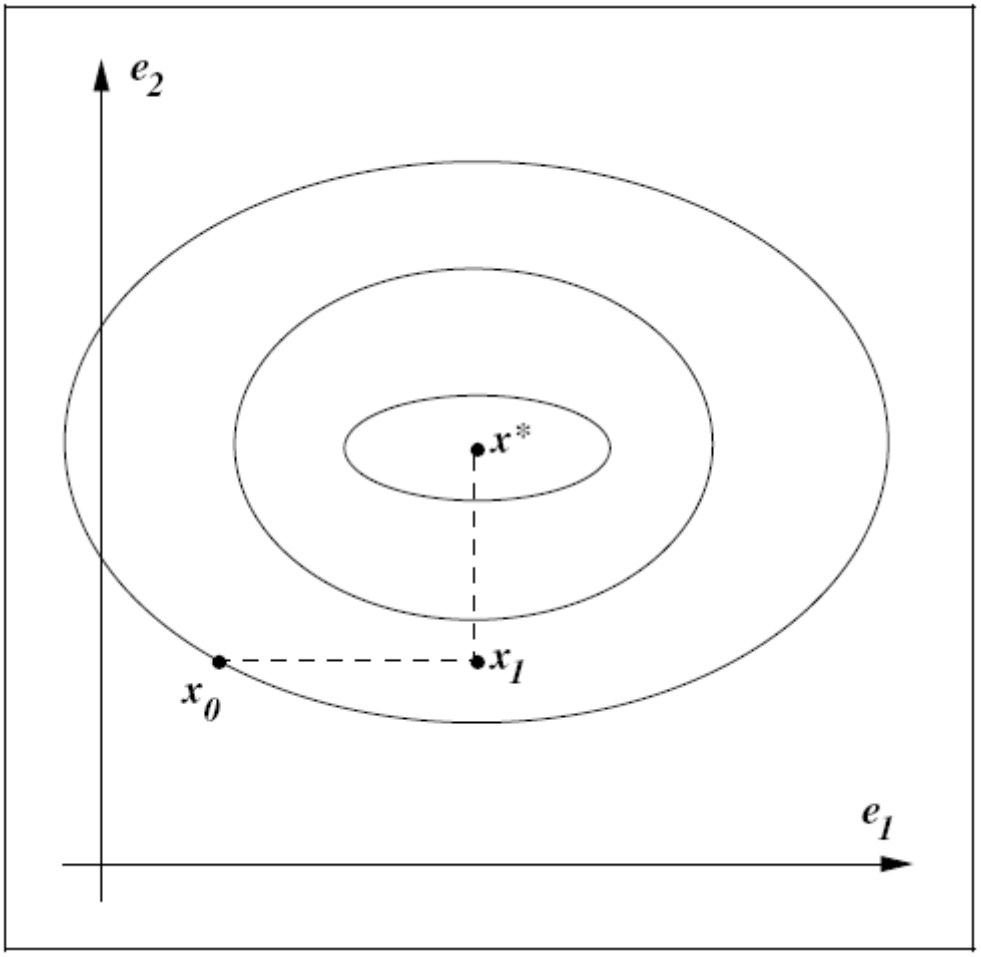
$$p_k^T A(x_k - x_0) = 0,$$

te da je

$$p_k^T A(x^* - x_0) = p_k^T A(x^* - x_k) = p_k^T (b - Ax_k) = -p_k^T r_k .$$

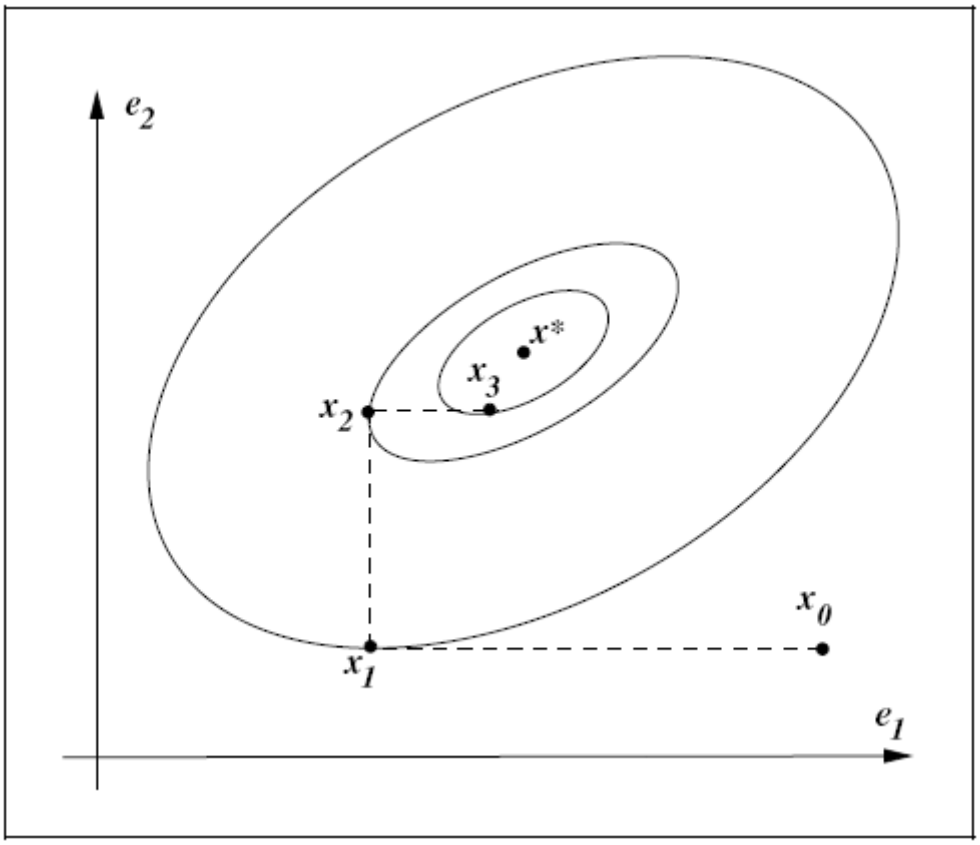
Uspoređujući ovu relaciju s (1.7) i (1.8) slijedi da je $\sigma_k = \alpha_k$. **Q.E.D.**

Postoji jednostavana interpretacija konjugiranih smjerova. Ako je matrica A u (1.2) dijagonalna, nivo skupovi funkcije $\phi(\cdot)$ su elipse čije su osi paralelne s osima koordinatnog sustava kao što se vidi na *slici (1.1).* Minimum ove funkcije možemo pronaći provodeći jednodimenzionalnu minimizaciju duž koordinatnih smjerova $e_1, e_2, ..., e_n$.



Slika (1.1) Sukcesivna minimizacija duž koordinatnih osi nalazi minimum kvadratne funkcije s dijagonalnim Hessianom u *n* iteracija.

Ukoliko A nije dijagonalna matrica, nivo skupovi funkcije $\phi(\cdot)$ su i dalje eliptični, ali najčešće nisu paralelni s koordinatnim osima. Strategija sukcesivnih minimizacija duž tih pravaca ne vodi do rješenja u *n* iteracija (čak niti u konačnom broju iteracija). Ova pojava je ilustrirana u dvodimenzionalnom primjeru na *slici (1.2).*



Slika (1.2) Sukcesivna minimizacija duž koordinatnih osi ne nalazi rješenje u *n* iteracija za opću konveksnu kvadratnu funkciju.

No mi možemo doći do lijepog ponašanja kao na *slici (1.2)* ako transformiramo problem tako da matricu *A* prvo dijagonaliziramo, te je onda minimiziramo duž koordinatnih osi. Problem ćemo transformirati tako da definiramo novu varijablu $\hat{x}$ kao

$$\hat{x} = S^{-1}x\,, \tag{1.9}$$

gdje je S $n \times n$ dimenzionalna matrica definirana sa $S = [p_0\, p_1 \ldots p_{n-1}]$, pri čemu je $\{p_0, p_1, \ldots, p_{n-1}\}$ skup konjugiranih vektora s obzirom na matricu A. Kvadratna funkcija $\phi(\cdot)$ definirana s (1.2) sada postaje

$$\hat{\phi}(\hat{x}) \overset{def}{=} \phi(S\hat{x}) = \frac{1}{2}\hat{x}^T(S^T A S)\hat{x} - (S^T b)^T\hat{x}\,.$$

Po svojstvu konjugacije (1.5) matrica $S^T AS$ je dijagonalna, pa možemo minimizirati vrijednost funkcije $\hat{\phi}$ izvodeći *n* jednodimenzionalnih minimizacija $\hat{x}$ duž koordinatnih osi. Traženje koordinata ovim načinom je ekvivalentno algoritmu konjugiranih smjerova (1.6) i (1.7) pa zaključujemo, kao i u teoremu (1.1), da algoritam konjugiranih vektora završava u najviše *n* koraka. Gledajući *sliku (1.1)* primjećujemo još jedno zanimljivo svojstvo. Kada je Hessian dijagonalna matrica, svaka koordinata minimizacije točno određuje jednu od komponenti rješenja *x**. Drugim riječima, nakon *k* jednodimenzionalnih minimizacija kvadratna funkcija je minimizirana na potprostoru razapetom s $e_1, e_2, ..., e_k$. Sljedeći teorem dokazuje ovaj bitan rezultat u općem slučaju u kojem Hessian kvadratne forme nije nužno dijagonalan. Za dokaz ovog rezultata koristit ćemo sljedeći izraz, koji slijedi iz relacija (1.4) i (1.6):

$$r_{k+1} = r_k + \alpha_k A p_k . \tag{1.10}$$

Koristit ćemo i oznaku $span\{p_0, p_1, ..., p_k\}$, koja označava skup svih linearnih kombinacija vektora $p_0, p_1, ..., p_k$.

TEOREM 1.2

Neka je $x_0 \in R^n$ početna točka i pretpostavimo da je niz $\{x_k\}$ generiran algoritmom konjugiranih smjerova (1.6) i (1.7), *k = 0,1,...,n*. Tada vrijedi

$$r_k^T p_i = 0 \qquad \text{za } i = 0,1,...,k-1, \tag{1.11}$$

i x_k je točka minimuma funkcije $\phi(x) = \frac{1}{2} x^T Ax - b^T x$ na skupu

$$\{x \mid x = x_0 + span\{p_0, p_1, ..., p_{k-1}\}\}. \tag{1.12}$$

Dokaz: Kao prvo, želimo pokazati da točka $\tilde{x}$ minimizira ϕ na skupu (1.12) ako i samo ako je $r(\tilde{x})^T p_i = 0$ za svaki $i = 0,1,...,k-1$. Definirajmo $h(\sigma) = \phi(x_0 + \sigma_0 p_0 + ... + \sigma_{k-1} p_{k-1})$, gdje je $\sigma = (\sigma_0, \sigma_1, ..., \sigma_{k-1})^T$. Kako je $h(\sigma)$ strogo konveksna i kvadratna funkcija, to slijedi da ima jedinstveni minimum σ^* koji zadovoljava

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

Po ovom lančanom pravilu, ova jednadžba implicira da je

$$\nabla \phi(x_0 + \sigma_0^* p_0 + ... + \sigma_{k-1}^* p_{k-1})^T p_i = 0, \qquad i = 0,1,...,k-1.$$

Pozivajući se na definiciju (1.3) imamo da je točka minimuma $\tilde{x} = x_0 + \sigma_0^* p_0 + ... + \sigma_{k-1}^* p_{k-1}$ na skupu (1.12) $r(\tilde{x})^T p_i = 0$, što i tvrdimo.
Sada ćemo indukcijom pokazati da x_k zadovoljava (1.11). Za slučaj kad je $k = 1$, imamo iz činjenice da $x_1 = x_0 + \alpha_0 p_0$ minimizira σ duž p_0, da je $r_1^T p_0 = 0$. Sada ćemo pokazati pretpostavku indukcije, konkretno, da je $r_{k-1}^T p_i = 0$ za $i = 0,1,...,k-2$. Zbog (1.10) imamo

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

pa je

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

iz definicije α_{k-1} (1.7). Međutim, za ostale vektore p_i, $i = 0,1,...,k-2$, imamo da je

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

gdje je $p_i^T r_{k-1} = 0$ zbog pretpostavke indukcije i $p_i^T A p_{k-1} = 0$ zbog konjugacije vektora p_i. Pokazali samo da je $r_k^T p_i = 0$ za $i = 0,1,...,k-1$ čime je dokaz i dovršen. **Q.E.D.**

Činjenicu da je trenutni rezidual r_k ortogonalan svim prijašnjim smjerovima često ćemo koristiti u ovom odjeljku.
Diskusija je do sada bila općenita, u tom što se odnosila na metodu konjugiranih smjerova (1.6) i (1.7) baziranu na bilo kojem izboru konjugiranih smjerova $\{p_0, p_1, ..., p_{n-1}\}$. Ima mnogo načina odabira skupa konjugiranih smjerova. Na primjer, svojstveni vektori $v_1, v_2, ..., v_n$ matrice A su međusobno ortogonalni kao i konjugirani s obzirom na matricu A, pa mogu biti upotrebljeni kao vektori $\{p_0, p_1, ..., p_{n-1}\}$. U primjeni, računanje cijelog skupa svojstvenih vektora velikih matrica zahtijeva previše računanja. Alternativni pristup je da matricu modificiramo Gram – Schmidtovim postupkom ortogonalizacije pa da napravimo skup konjugiranih smjerova radije nego ortogonalnih smjerova. (Ovo je lako za sprovesti jer su svojstva konjugacije i ortogonalizacije jako slična u svojoj biti). No trebamo uzeti u obzir da je i Gram – Schmidtov postupak također preskup, jer zahtijeva pohranjivanje cijelog skupa smjerova.

OSNOVNA SVOJSTVA METODE KONJUGIRANIH GRADIJENATA

Metoda konjugiranih gradijenata je metoda konjugiranih smjerova s jednim posebnim svojstvom: iz generiranog skupa konjugiranih vektora može se izračunati novi vektor p_k koristeći samo prethodni vektor p_{k-1}. Ne trebamo znati sve prethodne elemente $p_0, p_1, ..., p_{k-2}$ konjugiranog skupa; p_k je automatski konjugiran ovim vektorima. Ovo

izuzetno svojstvo implicira da ova metoda zahtijeva jako malo mjesta za pohranu podataka i računanje. U metodi konjugiranih gradijenata svaki smjer p_k je odabran tako da bude linearna kombinacija negativnog reziduala - r_k (koji je, po (1.3), smjer najbržeg silaska funkcije ϕ) i prethodnog smjera p_{k-1}. Pišemo

$$p_k = -r_k + \beta_k p_{k-1},$$

(1.13)

gdje je skalar β_k takav da p_{k-1} i p_k moraju biti konjugirani s obzirom na A. Množeći (1.13) s $p_{k-1}^T A$, uz uvjet da je $p_{k-1}^T A p_k = 0$ dobivamo da je

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

Odabiremo prvi pronađeni smjer p_0 da bude smjer najbržeg silaska početne točke x_0. Kao i u općoj metodi konjugiranih smjerova, izvodimo sukcesivnu jednodimenzionalnu minimizaciju duž svakog od pronađenih smjerova. Time smo specificirali kompletni algoritam kojeg zapisujemo ovako:

ALGORITAM 1.1 (KG-preliminarna verzija)

Zadan x_0;
Stavimo $r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0;$
dok $r_k \neq 0$ {

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k}; \tag{1.14a}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{1.14b}$$

$$r_{k+1} \leftarrow Ax_{k+1} - b; \tag{1.14c}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k}; \tag{1.14d}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \tag{1.14e}$$

$$k \leftarrow k+1; \tag{1.14f}$$

}

kraj(dok)

Ova verzija je korisna za proučavanje osnovnih svojstava metode konjugiranih gradijenata, no pokazat ćemo i mnogo učinkovitiju verziju poslije. Prvo ćemo pokazati da su smjerovi $p_0, p_1, ..., p_{n-1}$ konjugirani, što po Teoremu 1.1 implicira zaustavljanje u *n* koraka. Teorem ispod ustanovljuje to svojstvo, te i dva druga svojstva. Prvo, reziduali r_i su međusobno ortogonalni. Drugo, svaki smjer p_k i rezidual r_k je sadržan u Krylovljevu potprostoru stupnja *k* za r_0, definiranom kao

$$\kappa(r_0; k) \stackrel{def}{=} span\{r_0, Ar_0, ..., A^k r_0\} . \tag{1.15}$$

TEOREM1.3

Pretpostavimo da *k*-ta iteracija generirana metodom konjugiranih gradijenata nije rješenje (točka *x**). Tada vrijede sljedeća četiri svojstva:

$$r_k^T r_i = 0, \qquad \text{za } i = 0,1,...,k-1, \tag{1.16}$$

$$span\{r_0, r_1, ..., r_k\} = span\{r_0, Ar_0, ..., A^k r_0\}, \tag{1.17}$$

$$span\{p_0, p_1, ..., p_k\} = span\{r_0, Ar_0, ..., A^k r_0\}, \tag{1.18}$$

$$p_k^T A p_i = 0, \qquad \text{za } i = 0,1,...,k-1. \tag{1.19}$$

Stoga iteracije $\{x_k\}$ konvergiraju k *x** u najviše *n* koraka.

Dokaz: Dokaz provodimo indukcijom. Tvrdnje (1.17) i (1.18) su trivijalno ispunjene za *k = 0*, te (1.19) za *k* = 1. Pretpostavka indukcije je da su ova tri izraza istinita za neki *k*, a cilj je pokazati da su istinita i za *k* + 1.
Da bismo dokazali (1.17) prvo ćemo pokazati da je skup s lijeve strane jednakosti sadržan u skupu s desne strane jednakosti. Zbog pretpostavke indukcije, imamo iz (1.17) i (1.18) da

$$r_k \in span\{r_0, Ar_0, ..., A^k r_0\} \quad \text{i} \quad p_k \in span\{r_0, Ar_0, ..., A^k r_0\}$$

pa množeći drugi izraz s *A*, dobivamo da je

$$Ap_k \in span\{Ar_0, ..., A^{k+1} r_0\}. \tag{1.20}$$

Primijenimo li (1.10), dobivamo da je $r_{k+1} \in span\{r_0, Ar_0, ..., A^{k+1} r_0\}$. Kombinacija ovag izraza s pretpostavkom indukcije za (1.17) nas dovodi do zaključka da je

$$span\{r_0, r_1, ..., r_k, r_{k+1}\} = span\{r_0, Ar_0, ..., A^{k+1} r_0\} .$$

Kako bismo potvrdili da vrijedi i obrnuta inkluzija, koristit ćemo pretpostavku indukcije za (1.18), da bismo dobili da je

$$A^{k+1}r_0 = A(A^k r_0) \in span\{Ap_0, Ap_1, ..., Ap_k\}.$$

Zbog (1.10) imamo $Ap_i = (r_{i+1} - r_i)/\alpha_i$ za $i = 0,1,...,k$, pa slijedi da je

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

Dodamo li ovaj izraz pretpostavci indukcije za (1.17), zaključujemo da je

$$span\{r_0, Ar_0, ..., A^{k+1}r_0\} \subseteq span\{r_0, r_1, ..., r_k, r_{k+1}\}.$$

Stoga, relacija (1.17) vrijedi i kada *k* zamijenimo sa *k* + 1, kao što smo i tvrdili.
Pokazat ćemo da i (1.18) vrijedi kada *k* zamijenimo s *k* + 1 sljedećim argumentom

$$\begin{aligned}
&span\{p_0, p_1, ..., p_k, p_{k+1}\} \\
&= span\{p_0, p_1, ..., p_k, r_{k+1}\} && \text{po (1.14e)} \\
&= span\{r_0, Ar_0, ..., A^k r_0, r_{k+1}\} && \text{po pretpostavci indukcije za (1.18)} \\
&= span\{r_0, r_1, ..., r_k, r_{k+1}\} && \text{po (1.17)} \\
&= span\{r_0, Ar_0, ..., A^{k+1}r_0\} && \text{po (1.17) za } k+1.
\end{aligned}$$

Sada ćemo provjeriti uvjet konjugacije (1.19) zamijenivši *k* s *k* + 1. Množeći (1.14e) s Ap_i, $i = 0,1,...,k$, dobivamo

$$p_{k+1}^T Ap_i = -r_{k+1}^T Ap_i + \beta_{k+1} p_k^T Ap_i. \qquad (1.21)$$

Po definiciji β_k (1.14d), desna strana (1.21) nestaje za $i = k$. Za $i \le k-1$ trebamo uzeti u obzir nekoliko primjedbi. Prvo, primijetimo da naša pretpostavaka indukcije za (1.19) povlači da su smjerovi $p_0, p_1, ..., p_k$ konjugirani, tako da možemo po teoremu (1.2) zaključiti da je

$$r_{k+1}^T p_i = 0, \qquad \text{za } i = 0,1,...,k. \qquad (1.22)$$

Kao drugo, opetovanom primjenom (1.18), dobivamo da za $i = 0,1,...,k-1$ vrijedi sljedeća inkluzija

$$Ap_i \in Aspan\{r_0, Ar_0, ..., A^i r_0\} \subseteq span\{Ar_0, A^2 r_0, ..., A^{i+1}r_0\} \subseteq span\{p_0, p_1, ..., p_{i+1}\}. \qquad (1.23)$$

Iz (1.22) i (1.23) slijedi da

$$r_{k+1}^T A p_i = 0 \qquad \text{za } i = 0,1,...,k-1,$$

tako da prvi pribrojnik na desnoj strani u (1.21) nestaje za $i = 0,1,...,k-1$. Zbog pretpostavke indukcije za (1.19), drugi pribrojnik također nestaje, pa vrijedi da je $p_{k+1}^T A p_i = 0$ za $i = 0,1,...,k$. Naravno, argument indukcije vrijedi i za (1.19) .
Zaključujemo da je skup smjerova generiran metodom konjugiranih gradijenata uistinu skup konjugiranih smjerova, pa nam Teorem 1.1 govori da algoritam završava u najviše *n* koraka.
Konačno, provjerimo (1.16) sa neinduktivnim argumentom. Zbog toga što je skup smjerova konjugiran, iz (1.11) slijedi da je $r_k^T p_i = 0$ za svaki $i = 0,1,...,k-1$ i za svaki $k = 0,1,...,n-1$. Uređivanjem jednakosti (1.14e) dobivamo da

$$p_i = -r_i + \beta_i p_{i-1}$$

za $r_i \in span\{p_i, p_{i-1}\}$ i za svaki $i = 0,1,...,k-1$. Zaključujemo da je $r_k^T r_i = 0$ za svaki $i = 0,1,...,k-1$. Za završetak dokaza, zapišimo da je $r_k^T r_0 = -r_k^T p_0 = 0$, po definiciji p_0 u algoritmu (1.1) i po (1.11). **Q.E.D.**

Dokaz ovog teorema leži na činjenici da je prvi smjer p_0 u biti smjer najbržeg silaska - r_0; Činjenica je da rezultat ne vrijedi za druge izbore p_0. Kako su gradijenti r_k međusobno ortogonalni, termin *metoda konjugiranih gradijenata* je pogrešan. Smjerovi silaska, a ne gradijenti, su konjugirani s obzirom na A.

PRAKTIČNA FORMA METODE KONJUGIRANIH GRADIJENATA

Možemo izvesti i malo praktičniji oblik metode konjugiranih gradijenata koristeći Teoreme 1.2 i 1.3. Prvo, možemo iskoristiti (1.14e) i (1.11) da zamijenimo formulu (1.14a) za α_k sa

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$$

Kao drugo imamo iz (1.10) da je $\alpha_k A p_k = r_{k+1} - r_k$, pa primjenom (1.14e) i (1.11) još jednom možemo pojednostavniti formulu za β_{k+1}:

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

Koristeći ove formule zajedno s (1.10), dobivamo sljedeći standardni oblik metode konjugiranih gradijenata.

ALGORITAM1.2 (KG)

Zadan x_0;
Stavimo $r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0;$
dok $r_k \neq 0$ {

$$\alpha_k \leftarrow \frac{r_k^T p_k}{p_k^T A p_k}; \tag{1.24a}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{1.24b}$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k; \tag{1.24c}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}; \tag{1.24d}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \tag{1.24e}$$

$$k \leftarrow k+1; \tag{1.24f}$$

}

kraj(dok)

Za bilo koju danu početnu točku u *algoritmu 1.2* nikad ne trebamo znati vektore x, r i p dulje od zadnje dvije iteracije.U skladu s tim, implementacija ovog algoritma prepisuje preko starih vrijednosti ovih vektora da bi uštedjela na mjestu za pohranu podataka. Najveća računska zadaća koja se izvodi na svakom koraku je računanje produkta matice i vektora Ap_k, računanje skalarnih produkata $p_k^T(Ap_k)$ i $r_{k+1}^T r_{k+1}$, te računanje tri sume vektora. Skalarni produkt i sumiranje vektora mogu biti izvedeni malim umnošcima *n* operacija pomičnog zareza, dok računanje produkta matrice i vektora ovisi, naravno, o veličini matrice *A*. Metoda konjugiranih gradijenata se preporuča jedino za rješavanje velikih problema, dok su Gaussove eliminacije, ili drugi algoritmi faktorizacija kao što je dekompozicija singularne vrijednosti, preporučljiviji za ostale probleme jer su manje osjetljivi na pogreške zaokruživanja. Za velike probleme, metoda konjugiranih gradijenata ima prednost da ne sprema koeficijente matrice i (u suprotnosti s tehnikama faktorizacije) ne puni nizove koeficijentima spremajući matricu. Drugo ključno svojstvo je to da se katkad KG metoda približi rješenju jako brzo, no to nam je tema sljedećeg poglavlja.

Primjer1.1

Riješimo sustav linearnih jednadžbi *Ax = b* pomoću *algoritma(1.2),* gdje su elementi matrice *A* zadani s $A_{i,j} = 1/(i+j+1)$, te je $b = (1,1,...,1)^T$ i $x_0 = 0$. Iteracije ćemo brojiti za različite dimenzije matrice A i dokle god rezidual ne bude manji od 10^{-6}.
Algoritam(1.2) implementiran u programskom jeziku Dev C/C++ se nalazi na CD-u na kraju ovog rada s uputama za upotrebu u dodatku.
Dobili smo sljedeći rezultat za različite dimenzije *n* matrice *A*:

n	5	20	100	1000
broj iteracija	6	66	840	4824

BRZINA KONVERGENCIJE

Vidjeli smo da u egzaktnoj aritmetici metoda konjugiranih gradijenata završava u najviše *n* iteracija. No kad distribucija svojstvenih vrijednosti matrice *A* ima određena pogodna svojstva, algoritam će prepoznati rješenje u puno manje od *n* iteracija. Za objasniti ovo svojstvo, koristit ćemo Teorem 1.2 da pokažemo da je algoritam (1.2) optimalan u određenom bitnom smislu.
Iz (1.24b) i (1.18) imamo da je

$$\begin{aligned} x_{k+1} &= x_0 + \alpha_0 p_0 + ... + \alpha_k p_k \\ &= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + ... + \gamma_k A^k r_0, \end{aligned} \tag{1.25}$$

za neke konstante γ_i. Sada ćemo definirati $P_k^*(\cdot)$ kao polinom stupnja *k* s koeficijentima $\gamma_0, \gamma_1, ..., \gamma_k$. P_k^* može uzeti bilo skalar bilo kvadratnu matricu za svoj argument. Za matrični argument A imamo

$$P_k^*(A) = \gamma_0 I + \gamma_1 A + ... + \gamma_k A^k,$$

što nam dopušta da izrazimo (1.25) kao

$$x_{k+1} = x_0 + P_k^*(A) r_0. \tag{1.26}$$

Sada ćemo pokazati da između svih mogućih metoda čijih je prvih *k* koraka restringirano na Krylovljev potprostor $\kappa(r_0;k) \overset{def}{=} span\{r_0, Ar_0, ..., A^k r_0\}$, algoritam (1.2) daje najbolji rezultat minimizacije udaljenosti do rješenja nakon *k* koraka, gdje je ta udaljenost mjerena težinskom normom $\| \ \|_A$ definiranom sa

$$\|z\|_A^2 = z^T A z \,. \tag{1.27}$$

Koristeći ovu normu, definiciju funkcije ϕ (1.2) i činjenicu da *x** minimizira ϕ, slijedi da je

$$\frac{1}{2}\|x - x^*\|_A^2 = \frac{1}{2}(x - x^*)^T A(x - x^*) = \phi(x) - \phi(x^*). \tag{1.28}$$

Teorem 1.2 kaže da x_{k+1} minimizira ϕ na skupu $x_0 + span\{r_0, p_1, ..., p_k\}$, koji je po (1.18) isto što i $x_0 + span\{r_0, Ar_0, ..., A^k r_0\}$. Iz (1.26) slijedi da polinom P_k^* rješava sljedeći problem, gdje je minimum uzet po prostoru svih mogućih polinoma stupnja *k* :

$$\min_{P_k} \|x_0 + P_k(A)r_0 - x^*\|_A \,. \tag{1.29}$$

Ovo svojstvo optimalnosti ćemo istražiti u ostatku odjeljka. Zbog

$$r_0 = Ax_0 - b = Ax_0 - Ax^* = A(x_0 - x^*),$$

imamo

$$x_{k+1} - x^* = x_0 + P_k^*(A)r_0 - x^* = \left[I + P_k^*(A)A\right](x_0 - x^*). \tag{1.30}$$

Neka su $0 < \lambda_1 \le \lambda_2 \le ... \le \lambda_n$ svojstvene vrijednosti matrice A, te neka su $v_0, v_1, ..., v_n$ odgovarajući ortonormirani svojstveni vektori, tako da je

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T \,.$$

Kako svojstveni vektori razapinju cijeli prostor R^n, možemo pisati

$$x_0 - x^* = \sum_{i=1}^{n} \xi_i v_i \tag{1.31}$$

za neke koeficijente ξ_i. Vrijedi i da je svaki svojstveni vektor matrice A ujedno i svojstveni vektor $P_k(A)$ za bilo koji polinom P_k. Za našu konkretnu matricu A i njene svojstvene vrijednosti λ_i i svojstvene vektore v_i imamo da je

$$P_k(A)v_i = P_k(\lambda_i)v_i, \qquad \text{za } i = 1,2,...,n.$$

Supstitucijom (1.31) u (1.30) imamo

$$x_{k+1} - x^* = \sum_{i=1}^{n} \left[1 + \lambda_i P_k^*(\lambda_i)\right] \xi_i v_i .$$

Koristeći činjenicu da je $\|z\|_A^2 = z^T Az = \sum_{i=1}^{n} \lambda_i (v_i^T z)^2$, dobivamo da je

$$\|x_{k+1} - x^*\|_A^2 = \sum_{i=1}^{n} \lambda_i \left[1 + \lambda_i P_k^*(\lambda_i)\right]^2 \xi_i^2 . \tag{1.32}$$

Kako je polinom P_k^*, generiran metodom konjugiranih gradijenata, optimalan s obzirom na ovu normu, slijedi da je

$$\|x_{k+1} - x^*\|_A^2 = \min_{P_k} \sum_{i=1}^{n} \lambda_i \left[1 + \lambda_i P_k^*(\lambda_i)\right]^2 \xi_i^2 .$$

Ako izlučimo najveći izraz $\left[1 + \lambda_i P_k^*(\lambda_i)\right]^2$ iz jednadžbe, dobivamo da je

$$\|x_{k+1} - x^*\|_A^2 \le \min_{P_k} \max_{1 \le i \le n} \left[1 + \lambda_i P_k^*(\lambda_i)\right]^2 \left(\sum_{j=1}^{n} \lambda_j \xi_j^2 \right)$$

$$= \min_{P_k} \max_{1 \le i \le n} \left[1 + \lambda_i P_k^*(\lambda_i)\right]^2 \|x_0 - x^*\|_A^2 , \tag{1.33}$$

gdje smo iskoristili činjenicu da je $\|x_0 - x^*\|_A^2 = \sum_{j=1}^{n} \lambda_j \xi_j^2$.

Izraz (1.33) dopušta nam da odredimo brzinu konvergencije metode konjugiranih gradijenata procjenjujući broj nenegativnih skalara

$$\min_{P_k} \max_{1 \le i \le n} \left[1 + \lambda_i P_k^*(\lambda_i)\right]^2 . \tag{1.34}$$

Drugim riječima, tražimo polinom P_k koji će smanjiti vrijednost ovoga izraza na što je manje moguće. U nekim praktičnim slučajevima, možemo naći ovaj polinom eksplicitno i izvesti neke interesantne zaključke o svojstvima metode konjugiranih gradijenata. Sljedeći rezultat daje takav primjer.

TEOREM1.4

Ako A ima samo *r* različitih svojstvenih vrijednosti, tada će algoritam metode konjugiranih gradijenata završiti rješenjem u najviše *r* iteracija.

Dokaz: Pretpostavimo da svojstvene vrijednosti $\lambda_1, \lambda_2, ..., \lambda_n$ uzimaju *r* različitih vrijednosti $\tau_1 < \tau_2 < ... < \tau_r$. Definiramo polinom $Q_r(\lambda)$ kao

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \cdots \tau_r}(\lambda - \tau_1)(\lambda - \tau_2)\cdots(\lambda - \tau_r),$$

pri čemu vrijedi da je $Q_r(\lambda_i) = 0$ za $i = 1,2,...,n$ i $Q_r(0) = 1$. Zaključujemo da je $Q_r(\lambda) - 1$ polinom stupnja *r* s korijenom $\lambda = 0$, pa je funkcija $\overline{P}_{r-1}$ definirana kao

$$\overline{P}_{r-1}(\lambda) = (Q_r(\lambda) - 1)/\lambda$$

polinom stupnja *r* - 1. Stavljajući *k* = *r* - 1 u (1.34), dobivamo

$$0 \le \min_{P_{r-1}} \max_{1 \le i \le n} \left[1 + \lambda_i P_{r-1}(\lambda_i) \right]^2 \le \max_{1 \le i \le n} \left[1 + \lambda_i \overline{P}_{r-1}(\lambda_i) \right]^2 = \max_{1 \le i \le n} Q_r^2(\lambda_i) = 0.$$

Konstanta u (1.34) je nula za vrijednost $k = r - 1$, pa suptitucijom u (1.33) dobivamo da je $\|x_r - x^*\|_A^2 = 0$, iz čega slijedi $x_r = x^*$ kao što smo i tvrdili.
Q.E.D.

TEOREM1.5

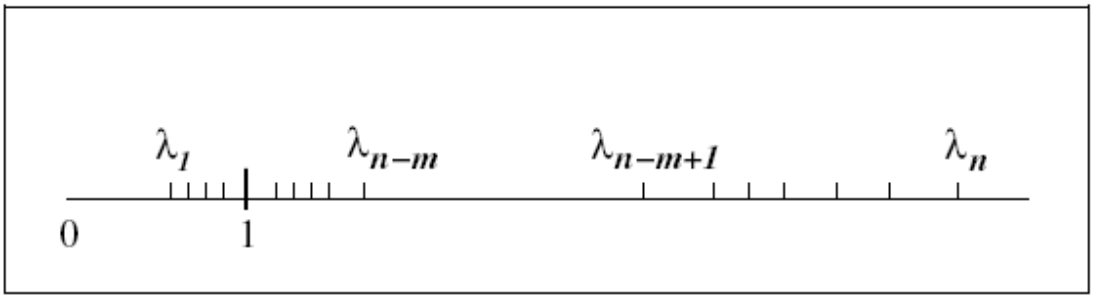
Ako A ima svojstvene vrijednosti $\lambda_1 \le \lambda_2 \le ... \le \lambda_n$, onda vrijedi sljedeća nejednakost

$$\|x_{k+1} - x^*\|_A^2 \le \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \|x_0 - x^*\|_A^2. \qquad (1.35)$$

Bez davanja detalja dokaza, opisat ćemo kako je ovaj rezultat sadržan u (1.33). Prvo, odabiremo polinom $\overline{P}_k$ stupnja *k* tako da polinom $Q_{k+1}(\lambda) = 1 + \lambda\overline{P}_k(\lambda)$ ima *k* najvećih svojstvenih vrijednosti $\lambda_n, \lambda_{n-1}, ..., \lambda_{n-k+1}$, kao i srednju vrijednost između λ_n i λ_{n-k}. Može se pokazati da je maksimalna vrijednost dosegnuta s Q_{k+1} točno $(\lambda_{n-k} - \lambda_1)/(\lambda_{n-k} + \lambda_1)$.
Sada ćemo ilustrirati kako Teorem 1.5 može biti korišten za predviđanje ponašanja metode konjugiranih gradijanata na sljedeći problem.
Pretpostavimo da imamo situaciju kao na *slici (1.3),* gdje se svojstvene vrijednosti matrice *A* sastoje od *m* velikih vrijednosti i ostatka od *n* - *m* malih svojstvenih vrijednosti grupiranih oko jednice. Ako definiramo $\varepsilon = \lambda_{n-m} - \lambda_1$, Teorem 1.5 nam kaže da nakon *m* + 1 koraka algoritma konjugiranih gradijenata, imamo

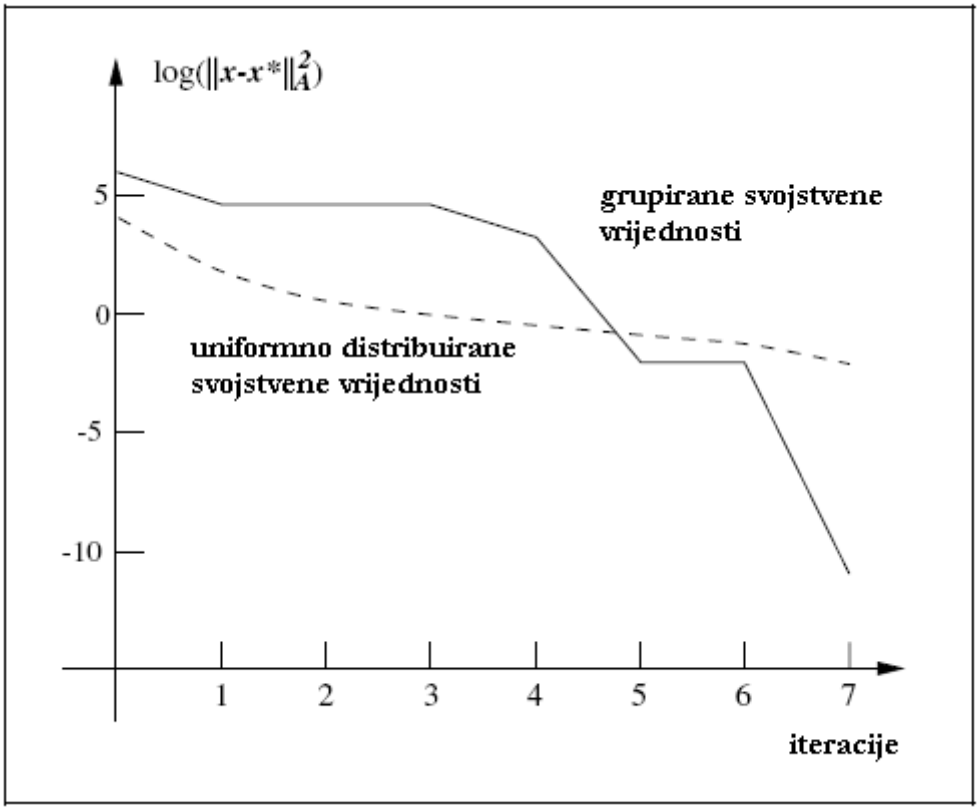
$$\|x_{m+1} - x^*\|_A \approx \varepsilon \|x_0 - x^*\|_A .$$

Za malu vrijednost ε, zaključujemo da će nam iteracije konjugiranih gradijenata omogućiti dobru procjenu rješenja nakon samo *m* + 1 koraka.



Slika(1.3) Dvije grupe svojstvenih vrijednosti

Slika (1.4) pokazuje nam ponašanje metode konjugiranih gradijenata na problemu ovog tipa, koji ima pet velikih svojstvenih vrijednosti sa svim malim svojstvenim vrijednostima smještenih između 0,95 i 1,05 te uspoređuje ovo ponašanje s ponašanjem metode konjugiranih gradijenata, čije su svojstvene vrijednosti nasumično raspoređene. U oba slučaja logaritmirat ćemo ϕ nakon svake iteracije.



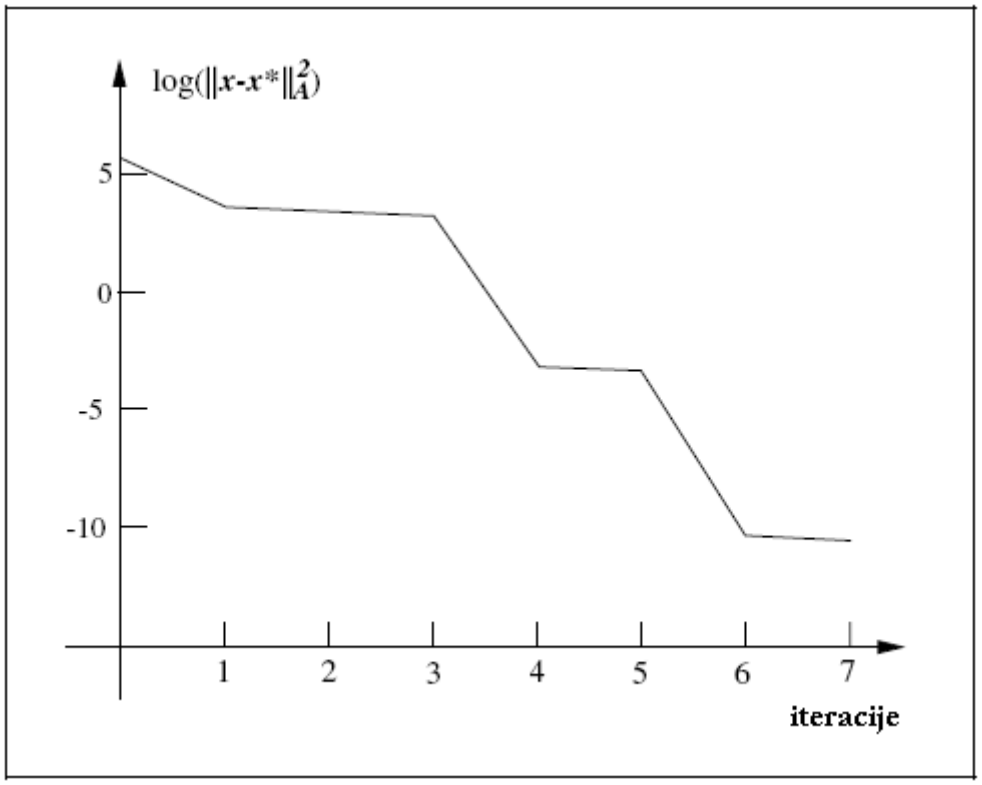
Slika(1.4) Primjena metode konjugiranih gradijenata na problem u kojem je pet svojstvenih vrijednosti velike a ostale su grupirane oko 1, te na matricu s uniformno raspoređenim svojstvenim vrijednostima.

Za problem s grupiranim svojstvenim vrijednostima *Teorem 1.5* predviđa oštar pad u procjeni pogrešaka u šestoj iteraciji. Primijetimo, međutim, da je ovaj pad postignut još jednu iteraciju ranije, ilistrirajući činjenicu da *Teorem 1.5* daje samo gornju granicu, te da brzina konvergencije može biti veća. Suprotnost vidimo u *slici (1.4),* gdje je za problem s nasumično raspoređenim svojstvenim vrijednostima brzina konvergencije manja i uniformnije raspoređena.

Slika (1.4) ilustrira još jednu zanimljivu karakteristiku: nakon svake iteracije(od ukupno sedam) problema s grupiranim svojstvenim vrijednostima, mjera pogreške oštro pada. Poopćenje argumenata koji vode do *Teorema 1.4* objašnjavaju ovo ponašanje. Možemo skoro sigurno reći da matrica *A* ima šest različitih svojstvenih vrijednosti: pet velikih svojstvenih vrijednosti i jedinicu. Tada bismo očekivali da mjera pogreške bude nula nakon šest iteracija. Zbog toga što su svojstvene vrijednosti oko jedan lagano raspršene, pogreška ne postaje jako mala do sedme iteracije.

Općenito vrijedi da, ako se svojstvene vrijednosti pojavljuju u *r* različitih grupa, iteracije metode konjugiranih gradijenata će aproksimativno riješiti problem u *r* koraka. Ovaj rezultat može biti provjeren konstruiranjem polinoma $\overline{P}_{r-1}$ tako da $1+\lambda\overline{P}_{r-1}(\lambda)$ ima nule unutar svake grupe. Ovaj polinom možda neće bit jednak nuli sa svojstvenim vrijednostima $\lambda_i, i = 1,2,...,n$, ali će njegova vrijednost biti mala kod ovih točaka, tako da će konstanta definirana u (1.34) biti mala za $k \geq r-1$. Ilustrirati ćemo ovo ponašanje *slikom (1.5),* koja pokazuje provođenje metode konjugiranjih gradijenata na matrici dimenzije *n* = 14 koja ima

četiri grupe svojstvenih vrijednosti: jednostruke svojstvene vrijednosti na 140 i 120, grupu od 10 svojstvenih vrijednosti blizu 10, te sa ostalim svojstvenim vrijednostima između 0,95 i 1,05. Nakon četiri iteracije, pogreška je značajno pala. Nakon šest iteracija, izdvojilo se rješenje dobre preciznosti.



Slika(1.5) Izvođenje metode konjugiranih gradijenata na matrici čije se svojstvene vrijednosti pojavljuju u četiri različite grupe.

Drugi, mnogo precizniji, izraz konvergencije za metodu konjugiranih gradijenata je baziran na uvjetovanosti matrice *A*, koju definiramo kao

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2 = \lambda_n / \lambda_1.$$

Vrijedi i da je

$$\|x_k - x*\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^k \|x_0 - x*\|_A. \qquad (1.36)$$

Ova ocjena često daje grube ocjene pogreške, ali može biti korisna u onim slučajevima gdje je jedina informacija koju imamo o A upravo procjena krajnjih svojstvenih vrijednosti λ_1 i λ_n. Ovo ograničenje bi trebalo biti uspređeno s metodom najbržeg silaska, koja u identičnoj formuli ovisi o uvjetovanosti $\kappa(A)$, a ne njezinom korijenu $\sqrt{\kappa(A)}$.

PREKONDICIONIRANJE

Metodu konjugiranih gradijenata možemo ubrzati ako transformiramo sustav linaernih jednadžbi tako da poboljšamo distribuciju svojstvenih vrijednosti matrice *A*. Ključni dio ovog procesa, poznatog kao prekondicioniranje, je promjena varijabli iz x u $\hat{x}$ duž nesingularne matrice *C*, tj.

$$\hat{x} = Cx\,. \tag{1.37}$$

Kvadratna funkcija ϕ definirana s (1.2) je transformirana u skladu s

$$\hat{\phi}(\hat{x}) = \frac{1}{2}\hat{x}^T (C^{-T}AC^{-1})\hat{x} - (C^{-T}b)^T \hat{x}\,. \tag{1.38}$$

Ako budemo koristili *algoritam (1.2)* da minimiziramo $\hat{\phi}$, ili ekvivalentno, da riješimo sustav linearnih jednadžbi

$$(C^{-T}AC^{-1})\hat{x} = C^{-T}b\,,$$

tada će brzina konvergencije ovisiti o svojstvenim vrijednostima matrice $C^{-T}AC^{-1}$ više nego onima matrice *A*. Stoga smo primorani odabrati *C* tako da svojstvene vrijednosti $C^{-T}AC^{-1}$ budu pogodnije za teoriju konvergencije o kojoj smo diskutirali u prošlom odjeljku. Možemo pokušati izabrati *C* tako da uvjetovanost $C^{-T}AC^{-1}$ bude mnogo manja nego originalna uvjetovanost matrice *A*, na primjer, tako da konstanta u (1.36) bude manja. Možemo također pokušati izabrati *C* tako da svojstvene vrijednosti $C^{-T}AC^{-1}$ budu grupirane, što nam, s obzirom na diskusiju u prethodnom odjeljku, osigurava da broj iteracija koji je potreban za pronalaženje dobrog rješenja, ne bude mnogo veći nego broj grupa.

Nije nužno do kraja eksplicitno završiti transformaciju (1.37). Radije, možemo primjeniti *algoritam (1.2)* na problem (1.38), u terminima varijable $\hat{x}$, te zatim invertirati transformaciju da ponovno izrazimo sve jednadžbe u terminima *x*. Ovaj proces izvođenja je *algoritam (1.3)* koji ćemo sada definirati. Događa se da *algoritam (1.3)* ne koristi *C* eksplicitno, nego radije koristi matricu $M = C^T C$ koja je simetrična i pozitivno definitna po konstrukciji.

ALGORITAM1.3 (prekondicionirani KG)

Zadan x_0, prekondicionarna matrica M;
Stavi $r_0 \leftarrow Ax_0 - b$;
Riješi $My_0 = r_0$ za y_0;
Stavi $p_0 = -y_0, k \leftarrow 0$;
dok $r_k \neq 0$ {

$$\alpha_k \leftarrow -\frac{r_k^T y_k}{p_k^T A p_k}; \tag{1.39a}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{1.39b}$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k; \tag{1.39c}$$

Riješi $My_{k+1} = r_{k+1}$; (1.39d)

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}; \tag{1.39e}$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k; \tag{1.39f}$$

$$k \leftarrow k+1; \tag{1.39g}$$

}

kraj(dok)

Ako stavimo da je *M* = I u *algoritmu (1.3)* ponovno dobijemo standardnu metodu konjugiranih gradijenata, tj. *algoritam (1.2)*.
Svojstva *algoritma (1.2)* generalizirana su naspram ovog slučaja u mnogim zanimljivim područjima. Konkretno, ortogonalno svojstvo (1.16) sukcesivnih reziduala postaje

$$r_i^T M^{-1} r_j = 0, \qquad \text{za svaki } i \neq j. \tag{1.40}$$

Glede računarskog truda, glavna razlika između prekondicionirane KG metode i neprekondicionirane KG metode je potreba da se riješe sustavi u oblika $My = r$ (korak 1.39d).

PRAKTIČNE PREKONDICIONARNE MATRICE

Niti jedna prekondicionarna strategija nije 'najbolja' za sve tipove matrica: što je bitnije između varijabilnih objekata – učinkovitost *M*, jeftinije računanje i pohranjivanje *M*, jeftinije rješenje $My = r$ – varira od problema do problema.

Dobra strategija prekondicioniranja je konstruirana za različite tipove matrica, posebno onih izvedenih iz diskretizacije parcijalnih diferencijalnih jednadžbi. Često je prekondicionarna matrica definirana na način da sustav $My = r$ teži pojednostavljenoj verziji orginalnog sistema $Ax = b$. Kao u mnogim drugim područjima optimizacije i numeričke analize, znanje o strukturi i podrijeklu problema (u ovom slučaju, znanje da je sustav $Ax = b$ konačno dimenzionalna reprezentacija parcijalne diferencijane jednadžbe), je ključ otkrivanja učinkovitih tehnika za rješavanje problema.

Primjer1.2

Riješimo isti problem kao i u primjeru(1.2), tj. sustav linearnih jednadžbi $Ax = b$ pomoću *algoritma(1.3)*, gdje su elementi matrice *A* zadani s $A_{i,j} = 1/(i + j + 1)$, te je $b = (1,1,...,1)^T$ i $x_0 = 0$. Iteracije ćemo brojiti za različite dimenzije matrice A i dokle god rezidual ne bude manji od 10^{-6}.

Algoritam(1.3) implementiran u programskom jeziku Dev C/C++ se nalazi na CD-u na kraju ovog rada s uputama za upotrebu u dodatku.

Novost u ovom primjeru je korištenje prekondicionarne matrice *M*. Bitno je da matrica *M* bude simetrična i pozitivno definitna pa smo uzeli dijagonalnu matricu gdje su elementi na dijagonali zadani s $M_{i,j} = 1 - 1/\sqrt{i+1}$.

Dobili smo sljedeći rezultat za različite dimenzije *n* matrice A:

n	5	20	100	1000
broj iteracija bez prekondicioniranja	6	66	840	4824
broj iteracija sa prekondicioniranjem	4	47	517	3861

Iz tablice vidimo da nam je uvođenje prekondicionarne matrice *M* rezultiralo i više no 20%-tnim poboljšanjem brzine algoritma, odnosno manjim brojem iteracija. Iako smo računali u svakoj iteraciji i sustav $My = r$, zbog manjeg broja iteracija, postigli smo bolji vremenski rezultat. Stoga se za velike probleme isplati uvesti nekakvu prekondicioniranu strategiju.

1.2 NELINEARNA METODA KONJUGIRANIH GRADIJENATA

Primjetili smo da *algoritam (1.2)* metode konjugiranih gradijenata može biti gledan kao algoritam minimizacije za konveksnu kvadratnu funkciju ϕ definiranu s (1.2). Prirodno je zapitati se da li možemo prilagoditi pristup minimizaciji konveksnih funkcija ili čak općenito nelinearne funkcije *f*. Ono što ćemo pokazati u ovom dijelu je da su i nelinearne metode konjugiranih gradijenata dobro proučene i pokazale su se poprilično uspješnima u praksi.

FLETCHER-REEVESOVA METODA

Fletcher i Reeves su pokazali kako proširiti metodu konjugiranih gradijenata na nelinearne funkcije tako da napravimo dvije jednostavne promjene u *algoritmu (1.2).* Prvo, na mjesto formule (1.24a) za korak duljine α_k (koji minimizira funkciju ϕ duž smjera silaska p_k), trebamo provesti jednodimenzionalnu optimizaciju koja identificira aproksimativni minimum nelinearne funkcije *f* duž p_k. Kao drugo, rezidual *r*, koji je jednostavno gradijent ϕ u *algoritmu (1.2),* mora biti zamijenjen s gradijentom nelinearne funkcije *f. O*ve promjene daju sljedeći algoritam nelinearne optimizacije.

ALGORITAM1.4 (FR)

Zadan x_0;
Procijeni $f_0 = f(x_0), \nabla f_0 = \nabla f(x_0)$;
Stavi $p_0 = -\nabla f_0, k \leftarrow 0$;
dok $\nabla f_k \neq 0$ {

Računaj α_k i postavi $x_{k+1} \leftarrow x_k + \alpha_k p_k$;

Procijeni ∇f_{k+1};

$$\beta_{k+1}^{FR} \leftarrow \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}; \tag{1.41a}$$

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k; \tag{1.41b}$$

$$k \leftarrow k+1; \tag{1.41c}$$

}

kraj(dok)

Ako odaberemo f da bude strogo konveksna kvadratna funkcija i α_k da bude egzaktni minimum, ovaj algoritam se reducira na linearnu metodu konjugiranih gradijenata, *algoritam (1.2). Algoritam (1.4)* se odnosi na velike nelinerane probleme optimizacije zato što svaka iteracija zahtijeva samo procjenu ciljne funkcije i njenog gradijenta. Matrične operacije su potrebne samo za računanje koraka i za samo nekoliko vektora su potrebna mjesta za pohranu. Za upotpuniti specifikacije *algoritma (1.4)* trebamo biti precizniji glede izbora jednodimenzionalne optimizacije parametra α_k. Zbog drugog pribrojnika u (1.41b) smjer silaska p_k može ne biti smjer silaska ukoliko α_k zadovoljava određene uvjete. Gledajući skalarni produkt u (1.41b) (gdje se k zamjenjuje s $k + 1$) i gradijent ∇f_k dobivamo da je

$$\nabla f_k^T p_k = -\|\nabla f_k\|^2 + \beta_k^{FR} \nabla f_k^T p_{k-1} \tag{1.42}$$

Ako je jednodimenzionalna optimizacija egzaktna, tako da je α_{k-1} lokalni minimum funkcije f duž smjera p_{k-1}, imamo da je $\nabla f_k^T p_{k-1} = 0$. U ovom slučaju imamo iz (1.42) da $\nabla f_k^T p_k < 0$, tako da je p_k uistinu smjer silaska. Ako jednodimenzionalna optimizacija nije egzaktna, svakako drugi pribrojnik u (1.42) može biti veći od prvog pribrojnika, pa možemo imati da je $\nabla f_k^T p_k > 0$, implicirajući da je p_k zapravo smjer rasta. Na sreću, mi možemo izbjeći tu situaciju zahtijevajući da duljina koraka α_k zadovoljava jake Wolfove uvjete, koji glase ovako:

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k, \tag{1.43a}$$

$$\left|\nabla f(x_k + \alpha_k p_k)^T p_k\right| \leq -c_2 \nabla f_k^T p_k, \tag{1.43b}$$

gdje je $0 < c_1 < c_2 < \frac{1}{2}$. Primjenjujući *Lemu 1.6* niže, možemo pokazati da uvjet (1.43b) implicira da je (1.42) negativan, pa tako zaključujemo da bilo koja tražena jednodimenzionalna optimizacija koja pridonosi α_k, koji zadovoljava (1.43), će osigurati da svi smjerovi p_k budu smjerovi silaska za funkciju f.

POLAK-RIBIEREOVA METODA I VARIJANTE

Postoji mnogo varijanti Fletcher-Reevesove metode koje se razlikuju većinom u izboru parametra β_k. Polak i Ribiere su uveli jednu takvu koja definira ovaj parametar kao

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2} \quad . \tag{1.44}$$

Pozivamo se na algoritam u kojem (1.44) zamjenjuje (1.41a) kao *algoritam PR.* Identičan je *algoritmu FR* kada je *f* strogo konveksna kvadratna funkcija i jednodimenzionalna optimizacija je egzaktna, budući da su po (1.16) gradijenti međusobno ortogonalni, pa je i $\beta_{k+1}^{PR} = \beta_{k+1}^{FR}$. Kada ovo primjenimo na općenite nelinearne funkcije s neegzaktnim jednodimenzionalnim optimizacijama, ponašanje ovih dvaju algoritama se znatno razlikuje. Numeričko iskustvo pokazuje da je *algoritam PR* mnogo robustniji i učinkovitiji. Iznenađujuće je da u *algoritmu PR* jaki Wolfeovi uvjeti ne moraju uvijek garantirati da je p_k smjer silaska. Ako definiramo parametar β kao

$$\beta_{k+1}^{+} = \max\{\beta_{k+1}^{PR}, 0\}, \tag{1.45}$$

dajući pojačanje algoritmu kojeg zovemo *algoritam PR+*, tada jednostavna prilagodba strogim Wolfeovim uvjetima osigurava da vrijedi svojstvo pada.
Ima mnogo drugih izbora za β_{k+1} koji se podudaraju s Fletcher-Reevesovom formulom β_{k+1}^{FR} u slučaju gdje je *f* kvadratna funkcija i jednodimenzionalna optimizacija je egzaktna. Hestenes-Stiefelova formula koja definira

$$\beta_{k+1}^{HS} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^T p_k} \tag{1.46}$$

daje poboljšanje algoritmu zvanom *algoritam HS,* koji je sličan *algoritmu PR*, kako u terminima teoretskih svojstava konvergencije, tako i u praktičnim provođenjima. Formula (1.46) može biti izvedena tako da se zahtjeva da uzastopni smjerovi silaska budu konjugirani s obzirom na Hessian na segmentu $[x_k, x_{k+1}]$ koji je definiran kao

$$\bar{G}_k \equiv \int_0^1 \nabla^2 f(x_k + \tau\alpha_k p_k)\, d\tau \, .$$

Pozivajući se na Taylorov teorem da je $\nabla f_{k+1} = \nabla f_k + \alpha_k \overline{G}_k p_k$, vidimo da za bilo koji smjer zadan u obliku $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$, uvjet $p_{k+1}^T \overline{G}_k p_k = 0$ zahtijeva da β_{k+1} bude dan s (1.46). Kasnije ćemo vidjeti da je moguće garantirati globalnu konvergenciju za bilo koji parametar β_k koji zadovoljava ograničenje

$$|\beta_k| \le \beta_k^{FR} \ , \tag{1.47}$$

za svaki $k \ge 2$. Ova činjenica sugerira sljedeću modifikaciju PR metode. Neka je za svaki $k \ge 2$

$$\beta_k = \begin{cases} -\beta_k^{FR} & ako \quad \beta_k^{PR} < -\beta_k^{FR} \\ \beta_k^{PR} & ako \quad |\beta_k^{PR}| \le \beta_k^{FR} \\ \beta_k^{FR} & ako \quad \beta_k^{PR} < \beta_k^{FR} \end{cases} \tag{1.48}$$

Druge varijante metode konjugiranih gradijanata su uvedene nedavno. Dva izbora za β_{k+1} koja imaju zanimljiva teoretska i računarska svojstva su

$$\beta_{k+1} = \frac{\|\nabla f_{k+1}\|^2}{(\nabla f_{k+1} - \nabla f_k)^T p_k} \tag{1.49}$$

i

$$\beta_{k+1} = \left(\hat{y}_k - 2 p_k \frac{\|\hat{y}_k\|^2}{\hat{y}_k^T p_k} \right)^T \frac{\nabla f_{k+1}}{\hat{y}_k^T p_k} \ \text{ sa } \ \hat{y}_k = \nabla f_{k+1} - \nabla f_k \tag{1.50}$$

Ova dva izbora garantiraju da je p_k smjer silaska, omogućujući da duljina koraka α_k zadovoljava Wolfeove uvjete. Algoritam metode konjugiranih gradijenata baziran na (1.49) i (1.50) je konkurentan Polak-Ribiereovoj metodi.

KVADRATNO ZAVRŠAVANJE I REINICIJALIZACIJA

Implementacija nelinearne metode konjugiranih gradijanata je usko povezana s linearnom metodom konjugiranih gradijenata. Obično, kvadratna (ili kubična) interpolacija duž smjera silaska p_k je ugrađena u proceduru jednodimenzionalne optimizacije. Ova karakteristika

garantira da kad je f strogo konveksna kvadratna funkcija, duljina koraka α_k je izabrana tako da bude točno jednodimenzionalni minimum. Slijedi da nelinearna metoda konjugiranih gradijenata reducira linearnu metodu konjugiranih gradijenata, *algoritam (1.2).*
Druga modifikacija koja je često korištena u nelinearnim procedurama konjugiranih gradijanata je reinicijalizacija. Provodimo je tako da nakon svakih *n* koraka stavimo da je $\beta_k = 0$ u (1.41a). Reinicijalizacija nam donosi periodično osvježenje algoritma, brišući stare informacije od kojih možda nema koristi. Možemo čak dokazati jaki teoretski rezultat oko reinicijalizacije; ono nas vodi do kvadratne konvergencije u *n* koraka:

$$\|x_{k+n} - x\| = O(\|x_k - x^*\|^2) . \tag{1.51}$$

Ovaj rezultat i nije toliko iznenađenje. Razmislimo o funkciji f koja je stogo konveksna kvadratna funkcija na okolini rješenja. Uzimajući u obzir da algoritam konvergira k rješenju, iteracije će kad tad upasti u kvadratno područje. U jednom trenutku algoritam će biti i reinicijaliziran na tom području, te od tog mjesta pa nadalje, njegovo ponašanje će biti čisto kao i u linearne metode konjugiranih gradijenata, *algoritam (1.2).* Konačni završetak će se pojaviti unutar *n* koraka od trenutka reinicijalizacije. Reinicijalizacija je bitno jer svojstvo završavanja u konačno mnogo koraka i druga svojstva *algoritma (1.2)* vrijede samo ako je početni smjer silaska p_0 jednak negativnom gradijentu.
Čak ako i funkcija f nije kvadratna na području rješenja, Taylorov teorem implicira da njena aproksimacija može biti kvadratna, omogućavajući joj glatkoću.
Iako je rezultat (1.51) interesantan i s teoretskog gledišta, ne mora biti relevantan u praktičnom kontekstu jer nelinearna metoda konjugiranih gradijenata može biti preporučena jedino za rješavanje problema sa velikim *n*. Reinicijalizacije se možda nikad neće pojaviti u takvim problemima, jer aproksimativno rješenje može biti locirano u manje od *n* koraka. Iako su nelinearne metode konjugiranih gradijenata ponekad implementirane bez reinicijalizacije, ili uključuju strategije za reinicijalizaciju koje uzimanju u obzir neke druge stvari osim broja iteracija. Najpopularnija strategija reinicijalizacije uzima u obzir tvrdnju (1.16) koja kaže da su gradijenti međusobno ortogonalni kada je f kvadratna funkcija. Reinicijalizacija se izvodi kad god su dva uzastopna gradijenta daleko od ortogonalnosti, koju mjeri test

$$\frac{\left|\nabla f_k^T \nabla f_{k-1}\right|}{\|\nabla f_k\|^2} \geq \nu , \tag{1.52}$$

gdje je tipična vrijednost za parametar ν jednaka 0.1.
Možemo također promatrati formulu (1.45) kao strategiju reinicijalizacije, jer će se p_{k+1} vratiti na smjer najbržeg silaska kad god je β_k^{PR} negativan. Suprotno testu (1.52), ove reinicijalizacije su poprilično rijetke jer je β_k^{PR} pozitivan većinu vremena.

PONAŠANJE FLETCHER-REEVESOVE METODE

Sada ćemo malo pomnije opisati *Fletcher-Reevesov algoritam (1.4)*, te dokazati da je globalno konvergentan i objasniti primijećene nedostatke.
Sljedeći rezultat daje uvjete na jednodimenzionalnu optimizaciju po kojoj su svi smjerovi upravo smjerovi silaska. Pretpostavimo da je skup $L = \{x : f(x) \le f(x_0)\}$ ograničen i da je f dvostruko neprekidno diferencijabilna, tako da imamo da postoji korak duljine α_k koji zadovoljava jake Wolfeove uvjete.

LEMA1.6

Pretpostavimo da je *algoritam (1.4)* implementiran s korakom duljine α_k koji zadovoljava jake Wolfeove uvjete (1.43) uz $0 < c_2 < \frac{1}{2}$. Tada metoda generira smjerove silaska p_k koji zadovoljavaju sljedeće nejednakosti:

$$-\frac{1}{1-c_2} \le \frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2} \le \frac{2c_2 - 1}{1-c_2} \qquad \text{za svaki } k = 0,1,.... \tag{1.53}$$

Dokaz: Primijetimo prvo da je funkcija $t(\xi) \overset{def}{=} (2\xi - 1)/(1-\xi)$ monotono rastuća na intervalu $\left[0, \frac{1}{2}\right]$ i da je $t(0) = -1$ i $t(\frac{1}{2}) = 0$. Zbog $c_2 \in (0, \frac{1}{2})$, imamo da je

$$-1 < \frac{2c_2 - 1}{1-c_2} < 0. \tag{1.54}$$

Uvjet silaska $\nabla f_k^T p_k < 0$ slijedi odmah čim ustanovimo (1.53).
Dokaz provodimo indukcijom. Za *k = 0* izraz u sredini u (1.53) je jednak -1, tako da koristeći (1.54), vidimo da su obje nejednakosti u (1.53) zadovoljene. Sljedeće, pretpostavimo da (1.53) vrijedi za neke $k \ge 1$. Iz (1.41b) i (1.41a) imamo

$$\frac{\nabla f_{k+1}^T p_{k+1}}{\|\nabla f_{k+1}\|^2} = -1 + \beta_{k+1} \frac{\nabla f_{k+1}^T p_k}{\|\nabla f_{k+1}\|^2} = -1 + \frac{\nabla f_{k+1}^T p_k}{\|\nabla f_k\|^2}.$$

(1.55)

Koristeći uvjet jednodimenzionalne optimizacije (1.43b) imamo da

$$\left|\nabla f_{k+1}^T p_k\right| \le -c_2 \nabla f_k^T p_k,$$

pa kombinirajući to s (1.55) i uzimajući u obzir (1.41a) dobivamo

$$-1+c_2\frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2}\le\frac{\nabla f_{k+1}^T p_{k+1}}{\|\nabla f_{k+1}\|^2}=-1-c_2\frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2}.$$

Supstitucijom izraza $\nabla f_k^T p_k \big/ \|\nabla f_k\|^2$ s lijeve strane hipoteze (1.53) dobivamo

$$-1-\frac{c_2}{1-c_2}\le\frac{\nabla f_{k+1}^T p_{k+1}}{\|\nabla f_{k+1}\|^2}\le-1+\frac{c_2}{1-c_2},$$

što pokazuje da (1.53) vrijedi i za $k + 1$. **Q.E.D.**

Ovaj rezultat koristi samo drugi jaki Wollfeov uvjet (1.43b); prvi Wolfeov uvjet (1.43a) će biti potreban u sljedećem dijelu da ustanovimo globalnu konvergenciju. Ocjene na $\nabla f_k^T p_k$ i (1.53) nameću granicu do koje norma koraka $\|p_k\|$ može rasti. Ovi faktori će odigrati presudnu ulogu u analizi konvergencije danoj ispod.
Lema 1.5 može biti također iskorištena za objašnjenje slabosti Fleetcher-Reevesove metode. Složit ćemo se da ako metoda generira loše smjerove i male korake, onda idući smjer i idući korak će vrlo vjerojatno opet biti loši. Neka θ_k označava kut između p_k i smjera najbržeg silaska $-\nabla f_k$ definiranog kao

$$\cos\theta_k=\frac{-\nabla f_k^T p_k}{\|\nabla f_k\|\|p_k\|} \tag{1.56}$$

Pretpostavimo da je p_k loš smjer silaska, u smislu da čini kut blizu 90° s $-\nabla f_k$, tj., $\cos\theta_k \approx 0$. Množeći obje strane (1.53) s $\|\nabla f_k\|/\|p_k\|$ i koristeći (1.56) dobivamo

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|}\le\cos\theta_k\le\frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|}, \qquad \text{za sve } k=0,1,.... \tag{1.57}$$

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

$$\|\nabla f_k\| << \|p_k\|.$$

Kako je p_k gotovo ortogonalan gradijentu, izgledno je da će biti mali korak od x_k do x_{k+1}, to jest, $x_{k+1} \approx x_k$. Ako je tako, onda imamo da je i $\nabla f_{k+1} \approx \nabla f_k$, pa je stoga

$$\beta_{k+1}^{FR} \approx 1, \tag{1.58}$$

po definiciji (1.41a). Koristeći ovu aproksimaciju i $\|\nabla f_{k+1}\| \approx \|\nabla f_k\| << \|p_k\|$, iz (1.41b) slijedi da je

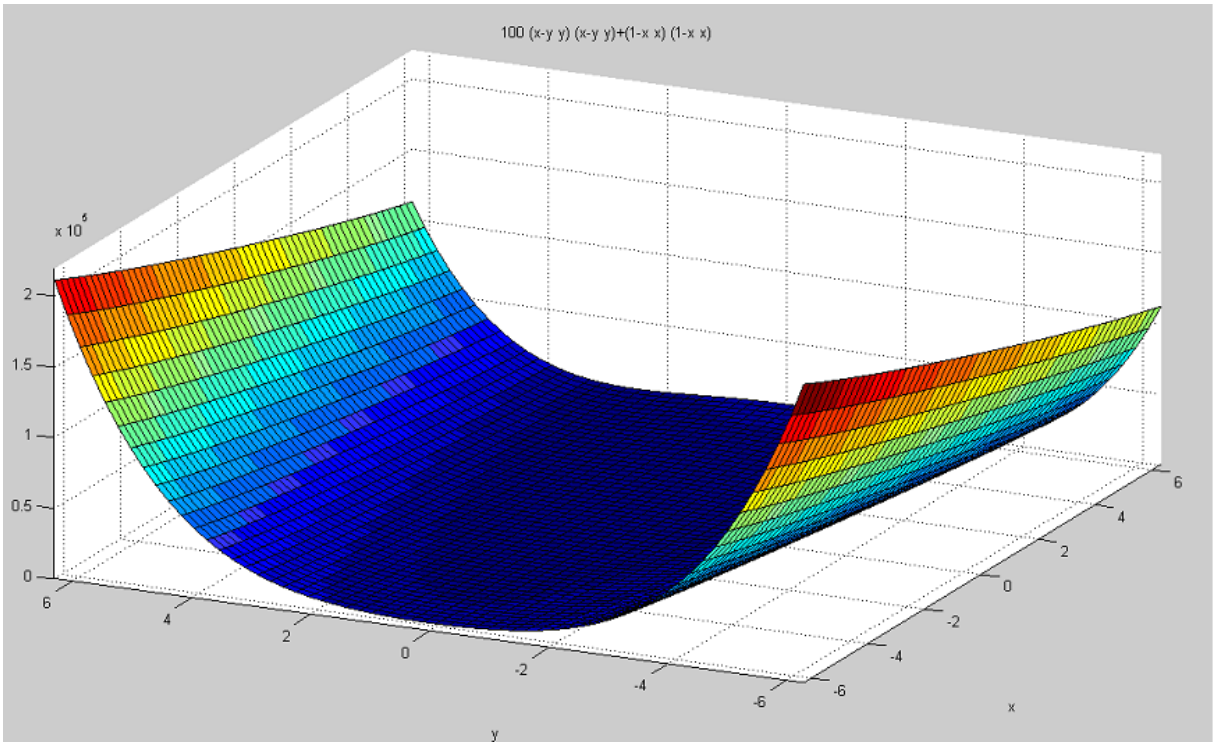
$$p_{k+1} \approx p_k ,$$

tako da novi smjer silaska poboljšava malo(ako imalo) prethodni. Slijedi da ako vrijedi $\cos\theta_k \approx 0$ kod neke *k*-te iteracije te ako je podintervalni korak mali, slijediti će dugi niz neproduktivnih iteracija.
Polak-Ribiereova metoda se ponaša poprilično drugačije u ovim uvjetima. Ako smjer silaska p_k zadovoljava $\cos\theta_k \approx 0$ za neki *k*, te ako je podintervalni korak mali, slijedi da ćemo supstitucijom $\nabla f_{k+1} \approx \nabla f_k$ u (1.44) dobiti da je $\beta_{k+1}^{PR} \approx 0$. Iz formule (1.41b) nalazimo da će novi smjer silaska p_{k+1} bit jako blizu smjeru najbržeg silaska $-\nabla f_{k+1}$, i $\cos\theta_{k+1}$ će biti blizu 1. Stoga *algoritam PR* u biti izvodi reinicijalizaciju kad naleti na loš smjer. Isti argument može biti primijenjen i na *algoritam PR+* i *HS*. Za FR-PR varijantu, definiranu s (1.48), imamo da je $\beta_{k+1}^{FR} \approx 1$ i $\beta_{k+1}^{PR} \approx 0$. Formula (1.48) stoga postavlja $\beta_{k+1} = \beta_{k+1}^{PR}$ kao što smo i željeli. Modifikacija (1.48) čini da se izbjegavaju neučinkovitosti FR metode, dok ne prolazi test globalne konvergencije na ovoj metodi.
Nepoželjno ponašanje FR metode unaprijed određeno argumentima od gore, može se promatrati i u praksi. Na primjer, problem gdje je *n = 100* u kojem je $\cos\theta_k$ reda 10^{-2} za stotine iteracija i koracima $\|x_k - x_{k-1}\|$ reda 10^{-2}. *Algoritam FR* zahtijeva tisuće iteracija za riješiti ovaj problem, dok *algoritam PR* zahtijeva samo 37 iteracija. U ovom primjeru FR metoda se izvodi puno bolje ako se periodično reinicijalizira duž smjera najbržeg silaska, jer svaka reinicijalizacija završava ciklus loših koraka. Općenito, *algoritam FR* ne bi trebao biti implementiran bez neke strategije reinicijalizacije.

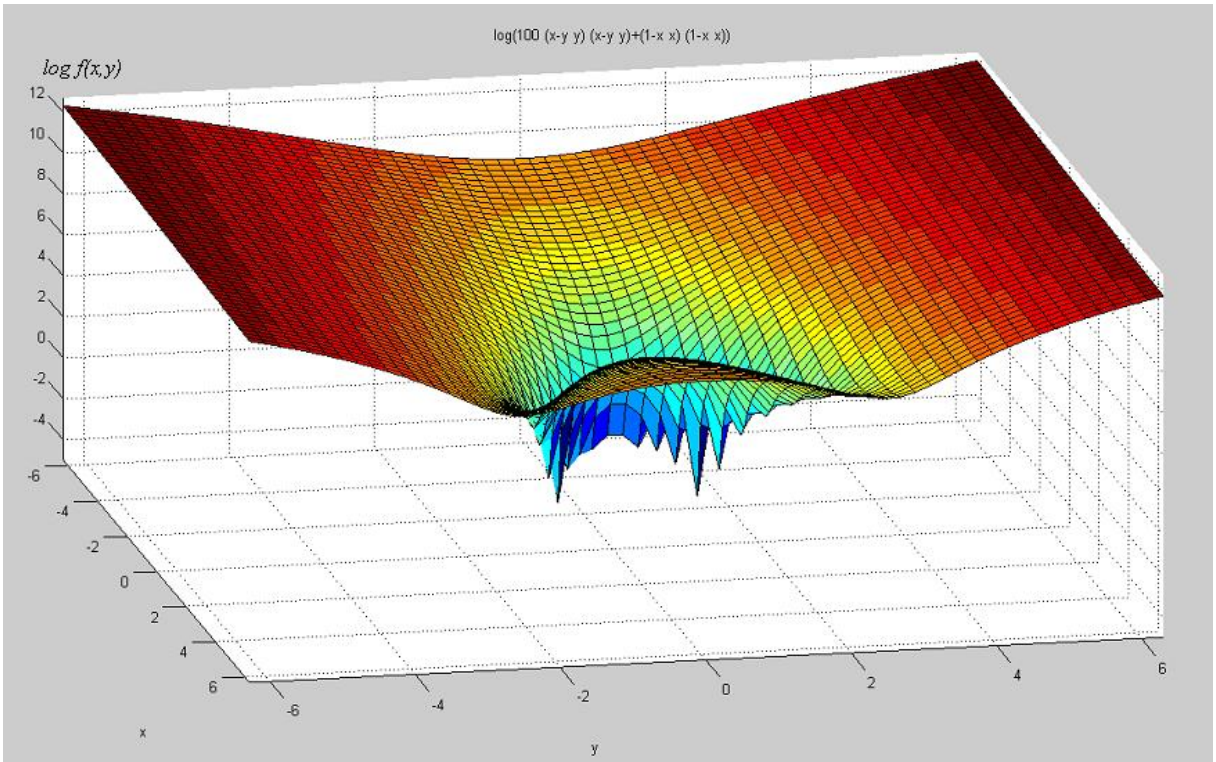
Primjer1.3

Testirat ćemo učinkovitost Fletcher-Reevesove i Polak-Ribiereove metode konjugiranih gradijenata na Rosenbrockovoj funkciji koja ima minimum u zakrivljenoj i vrlo sporo spuštajućoj „dolini“.
Algoritam(1.4) i algoritam (1.5) su implementirani u programskom jeziku Dev C/C++ i nalaze se na CD-u na kraju ovog rada s uputama za upotrebu u dodatku. Podaci dobiveni ovim algoritmima se nalaze u datotekama rosen1.txt i rosen2.txt, također na CD-u, te su isti korišteni za crtanje Rosenbrockove funkcije i put konvergencije obiju metoda u programskom jeziku MATLAB.
Obje metode kreću iz iste početne točke (0.25,0) i nalaze minimum u točki (1,1).

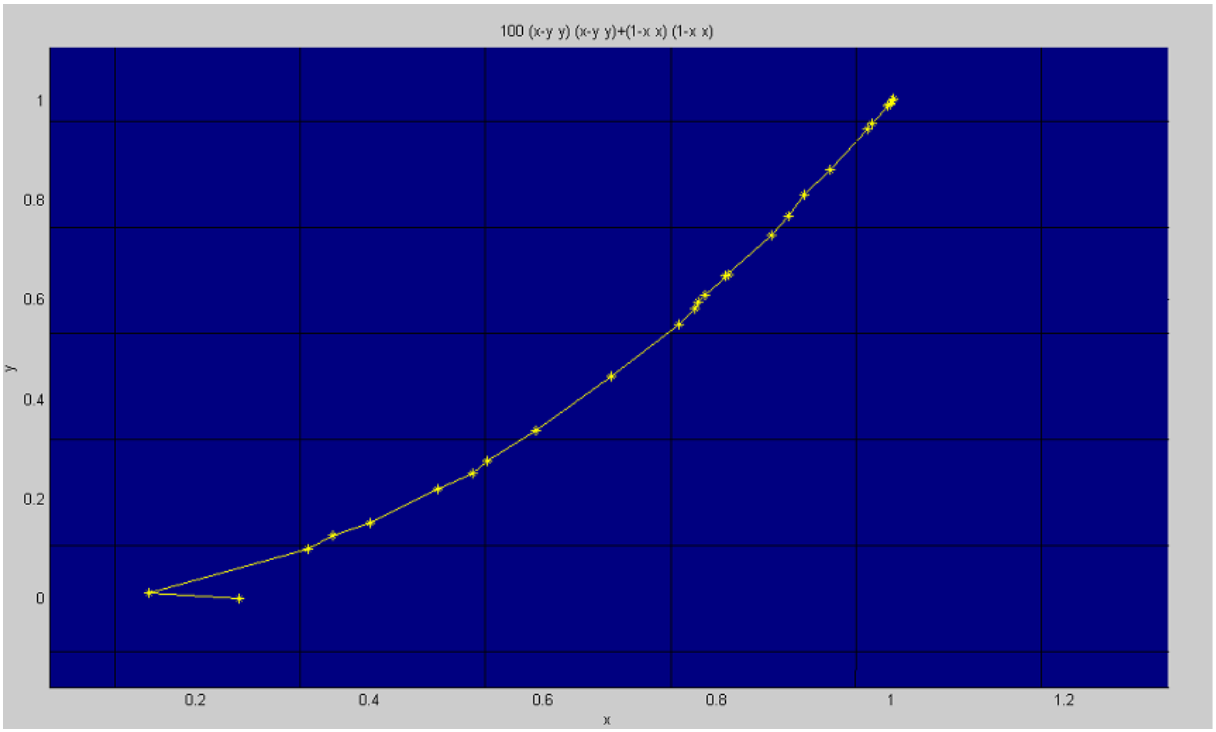
Rosenbrockova funkcija $f(x,y) = 100(x - y^2)^2 + (1 - x^2)^2$



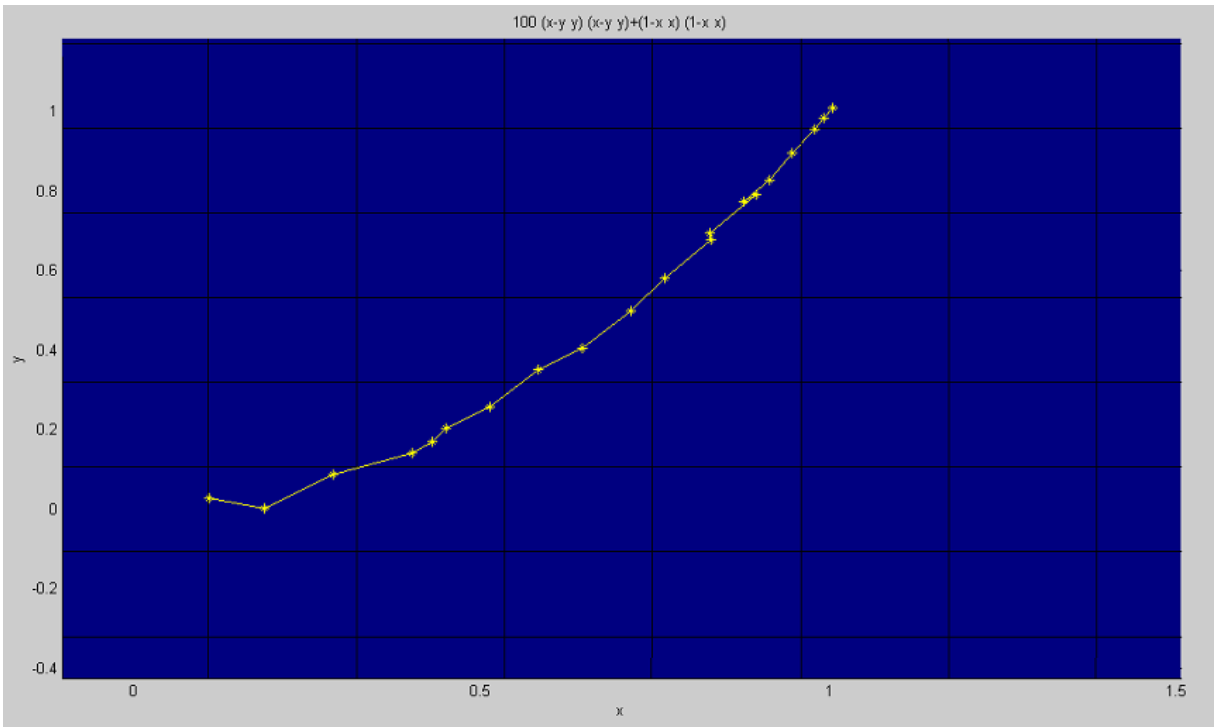
Rosenbrockovu funkciju ćemo i logaritmirati da se bolje vidi da ta velika dolina i nije tako „monotona“ kao što se čini na prvi pogled. Na ovoj slici se već i nadzire minimum.



Fletcher-Reevesova metoda nalazi minimum u točki (1,1) u 49 iteracija.



Polak-Ribiereova metoda nalazi minimum također u (1,1), ali je vidno „grublja“ od Fletcher-Reevesove metode.Minimum nalazi u 21 iteraciji.



GLOBALNA KONVERGENCIJA

Za razliku od metode konjugiranih gradijenata, čija svojstva konvergencije su dobro shvaćena i za koja se zna da su optimalna kao što smo gore i opisali, nelinearna metoda konjugiranih gradijenata pokazuje iznenađujuća svojstva konvergencije.
Prikazat ćemo neke glavne rezultate poznate za korištenje FR i PR metoda koristeći praktične jednodimenzionalne optimizacije.
Za svrhovitost ovog dijela, koristit ćemo sljedeće nerestriktivne pretpostavke na ciljnu funkciju.

PRETPOSTAVKE1.1

(i) Skup $L = \{x : f(x) \le f(x_0)\}$ je ograničen.

(ii) Na otvorenoj okolini *N* od L, ciljna funkcija f je Lipschitzova funkcija.

Ove pretpostavke impliciraju da postoji konstanta $\bar{\gamma}$ takva da je

$$\|\nabla f(x)\| \le \bar{\gamma}, \qquad \text{za svaki } x \in L. \tag{1.59}$$

Naše osnovno analitičko oruđe u ovom dijelu je Zouendijkov teorem, koji kaže da pod uvjetima (1.1) bilo koja jednodimenzionalna optimizacija oblika $x_{k+1} = x_k + \alpha_k p_k$, gdje je p_k smjer silaska, a α_k zadovoljava Wolfeove uvjete, daje ograničenje

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty. \tag{1.60}$$

Možemo iskoristiti ovaj rezultat da dokažemo globalnu konvergenciju algoritama koji se periodično reinicijaliziraju stavljanjem $\beta_k = 0$. Ako $k_1, k_2, itd.$ označavaju iteraciju na kojoj se reinicijalizacija pojavljuje, imamo iz (1.60) da

$$\sum_{k=k_1,k_2,\ldots}^{\infty} \|\nabla f_k\|^2 < \infty. \tag{1.61}$$

Ako dopustimo ne više od $\bar{n}$ iteracija između reinicijalizacija, niz $\{k_j\}_{j=1}^{\infty}$ je beskonačan i iz (1.61) imamo da je $\lim_{j\to\infty} \|\nabla f_{k_j}\| = 0$. To je podniz gradijenata koji se približava nuli, ili ekvivalentno

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0. \tag{1.62}$$

Ovaj rezultat se pojavljuje podjednako u reinicijaliziranim vezijama svih algoritama diskutiranim u ovom poglavlju.
Još više je interesantno proučavanje globalne konvergencije metoda konjugiranih gradijenata bez reinicijalizacije, zato što za velike probleme (recimo n>1000) očekujemo da nađemo rješenje u manje od *n* iteracija-prvo mjesto na kojem će regularna reinicijalizacija naći svoje mjesto. Naše proučavanje velikih nizova iteracija konjugiranih gradijenata bez reinicijalizacije otkriva neka iznenađujuća ponavljanja u njihovom ponašanju.
Možemo izgraditi na *Lemi 1.6* i Zoutendijkovom rezultatu (1.60) dokaz globalne konvergencije rješenja FR metode. Dok ne možemo pokazati da je limit niza gradijenata ∇f_k nula, sljedeći rezultat pokazuje da ovaj niz nije ograničen dalje od nule.

TEOREM1.7 (Al-Baali)

Pretpostavimo da vrijede pretpostavke (1.1) i da je algoritam (1.4) implementiran s jednodimenzionalnom optimizacijom koja zadovoljava jake Wolfeove uvjete (1.43), uz uvjet $0 < c_1 < c_2 < \frac{1}{2}$. Tada je

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0 . \tag{1.63}$$

Dokaz: Dokaz provodimo kontradikcijom. Pretpostavimo da vrijedi suprotno, tj., da postoji konstanta $\gamma > 0$ takva da je

$$\|\nabla f_k\| \geq \gamma , \tag{1.64}$$

za svaki dovoljno veliki *k*. Supstitucijom lijeve strane nejednakosti iz (1.57) u Zoutendijkovom uvjetu (1.60) dobivamo

$$\sum_{k=0}^{\infty} \frac{\|\nabla f_k\|^4}{\|p_k\|^2} < \infty . \tag{1.65}$$

Koristeći (1.43b) i (1.53) dobivamo

$$\left|\nabla f_k^T p_{k-1}\right| \leq -c_2 \nabla f_{k-1}^T p_{k-1} \leq \frac{c_2}{1-c_2}\|\nabla f_{k-1}\|^2 . \tag{1.66}$$

Iz (1.41b) i pozivanjem na definiciju β_k^{FR} (1.41a), dobivamo da je

$$\|p_k\|^2 \leq \|\nabla f_k\|^2 + 2\beta_k^{FR}\left|\nabla f_k^T p_{k-1}\right| + (\beta_k^{FR})^2\|p_{k-1}\|^2$$

$$\leq \|\nabla f_k\|^2 + \frac{2c_2}{1-c_2}\beta_k^{FR}\|\nabla f_{k-1}\|^2 + (\beta_k^{FR})^2\|p_{k-1}\|^2$$

$$= \frac{1+c_2}{1-c_2}\|\nabla f_k\|^2 + (\beta_k^{FR})^2\|p_{k-1}\|^2 .$$

Primjenjujući ovu relaciju opetovano i definirajući $c_3 \overset{def}{=} (1+c_2)/(1-c_2) \geq 1$ imamo

$$\begin{aligned}\|p_k\|^2 &\leq c_3\|\nabla f_k\|^2 + (\beta_k^{FR})^2(c_3\|\nabla f_{k-1}\|^2 + (\beta_{k-1}^{FR})^2(c_3\|\nabla f_{k-2}\|^2 + ... + (\beta_1^{FR})^2\|p_0\|^2))...) \\ &= c_3\|\nabla f_k\|^4 \sum_{j=0}^{k}\|\nabla f_j\|^{-2} ,\end{aligned} \tag{1.67}$$

gdje koristimo činjenicu da je

$$(\beta_k^{FR})^2(\beta_{k-1}^{FR})^2...(\beta_{k-i}^{FR})^2 = \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-i-1}\|^4}$$

i da je $p_0 = -\nabla f_0$. Iskoristimo li i ograničenja (1.59), (1.64) i (1.67) imamo da je

$$\|p_k\|^2 \leq \frac{c_3\bar{\gamma}^4}{\gamma^2}k , \tag{1.68}$$

što implicira da je

$$\sum_{k=1}^{\infty}\frac{1}{\|p_k\|^2} \geq \gamma_4\sum_{k=1}^{\infty}\frac{1}{k},$$

(1.69)

za neku pozitivnu konstantu γ_4.
S druge strane, iz (1.64) i (1.65) imamo da je

$$\sum_{k=1}^{\infty}\frac{1}{\|p_k\|^2} < \infty . \tag{1.70}$$

Ako kombiniramo ovu nejednakost s (1.69), dobivamo da je $\sum_{k=1}^{\infty}\frac{1}{k} < \infty$, što nije istina.Stoga (1.54) ne vrijedi i tvrdnja (1.63) je dokazana. **Q.E.D.**

Ovaj rezultat globalne konvergencije može biti proširen na bilo koji izbor β_k koji zadovoljava (1.47), pa posebno i FR-PR metode dane s (1.48).
Općenito možemo pokazati da postoje konstante $c_4, c_5 > 0$ takve da je

$$\cos\theta_k \geq c_4 \frac{\|\nabla f_k\|}{\|p_k\|}, \quad \frac{\|\nabla f_k\|}{\|p_k\|} \geq c_5 > 0, \qquad k = 1,2,...$$

pa iz (1.60) slijedi da je

$$\lim_{k\to\infty} \|\nabla f_k\| = 0.$$

Štoviše, ovaj rezultat može vrijediti za Polak-Ribiereovu metodu pod pretpostavkom da je *f* strogo konveksna i da je korištena ezgaktna jednodimenzionalna optimizacija.
Za nekonveksne funkcije, međutim, nije moguće dokazati rezultat kao što je *Teorem 1.7* za *algoritam PR*. Ova činjenica je neočekivana, budući da se Polak-Ribiereova metoda bolje izvodi u praksi nego Fletcher-Reevesova metoda. Sljedeći iznenađujući rezultat nam pokazuje da se PR metoda može beskonačno vrtjeti u krug-ciklirati bez približavanja rješenju, čak ako je i upotrebljena idealna jednodimenzionalna optimizacija.(Pod „idealna“ mislimo da jednodimenzionalna optimizaciju vraća α_k, koji je prva pozitivna stacionarna točka funkcije $t(\alpha)) f(x_k + \alpha p_k).)$

TEOREM1.8

Neka je dana Polak-Ribiereova metoda s idealnom jednodimenzionalnom optimizacijom. Tada postoji dvostruko neprekidna diferencijabilna ciljna funkcija $f : R^3 \to R$ i početna točka $x_0 \in R^3$, takva da je niz gradijenata $\|\nabla f_k\|$ ograničen oko nule.

Dokaz ovog teorema je poprilično složen. Demonstrira postojanje ciljne funkcije bez konstruiranja te iste eksplicitno. Rezultat je interesantan, budući da duljina koraka pretpostavljena u dokazu – prva stacionarna točka – može biti prihvaćena kod svih praktičnih algoritama jednodimenzionalne optimizacije koji su trenutno u upotrebi. Dokaz *Teorema 1.8* zahtjeva da neki uzastopni smjerovi silaska postanu gotovo smjerovi uzlaska. U slučaju idealne jednodimenzionalne optimizacije, ovo se događa jedino ako je $\beta_k < 0$, tako da analiza sugerira *algoritam PR+*, u kojem postavimo β_k na nulu kad god postane negativan.
Spomenuli smo ranije da strategija jednodimenzionalne optimizacije bazirana na malim modifikacijama Wolfeovih uvjeta garantira da svi smjerovi generirani *algoritmom PR+* budu i smjerovi silaska. Koristeći ove činjenice, moguće je dokazati globalnu konvergenciju kao u *Teoremu 1.7* za *algoritam PR+*. Atraktivno svojstvo formula (1.49) i (1.50) je da globalna konvergencija može biti ustanovljena bez modifikacija jednodimenzionalne optimizacije baziranih na Wolfeovim uvjetima.

NUMERIČKA IZVEDBA I PRIMJERI

Tablica (1.1) ilustrira izvedbu algoritama FR,PR i PR+ bez reinicijalizacije. Za ove testove, parametri iz Wolfeovih jakih uvjeta (1.43) su izabrani da budu $c_1 = 10^{-4}$ i $c_2 = 0.1$. Iteracije će biti zaustavljene kada

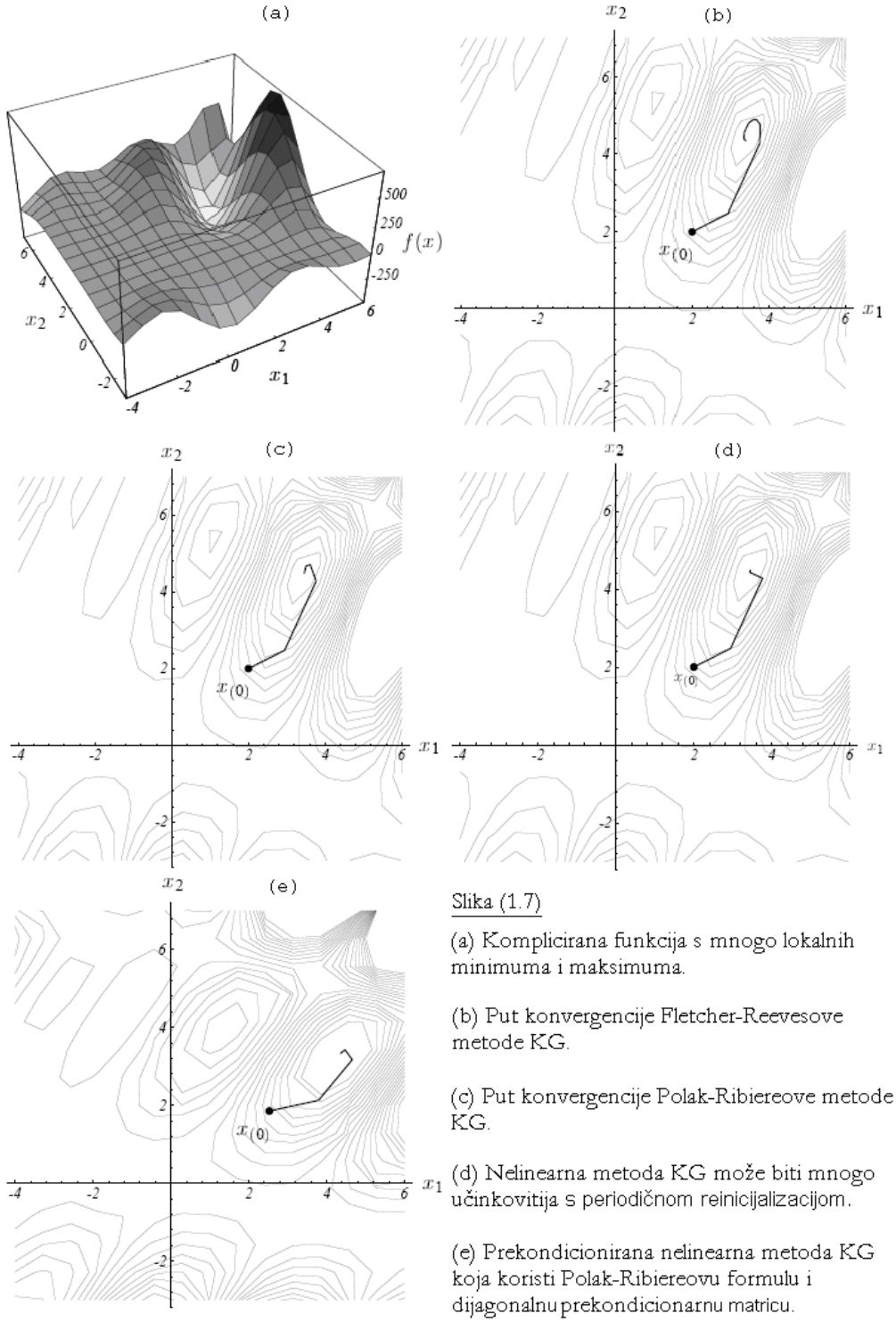
$$\|\nabla f_k\|_\infty < 10^{-5}(1+|f_k|).$$

Ako ovaj uvjet nije zadovoljen nakon 10000 iteracija, proglašavamo neuspjeh(označen sa * u tabeli).
Zadnji stupac, označen s „mod“ označava broj iteracija *algoritma PR+* za koji je prilagodba (1.45) bila potrebna da osigura da je $\beta_k^{PR} \geq 0$. *Algoritam FR* za problem GENROS uzima vrlo male korake dalje od rješenja koje vodi malim poboljšanjima u ciljnoj funkciji, te konvergencija nije postignuta unutar maksimalnog broja iteracija.
PR algoritam, ili njegova varijanta PR+, nisu uvijek učinkovitiji nego FR algoritam. Ima i laganu manu što mora spremiti jedan vektor više. No ipak, preporuča se da korisnici odaberu algoritam PR,PR+ ili PR-FR ili metodu baziranu na (1.49) i (1.50).

Problem	n	Alg FR it/f-g	Alg PR it/f-g	Alg PR+ it/f-g	mod
CALCVAR3	200	2808/5617	2631/5263	2631/5263	0
GENROS	500	*	1068/2151	1067/2149	1
XPOWSING	1000	533/1102	212/473	97/229	3
TRIDIA1	1000	264/531	262/527	262/527	0
MSQRT1	1000	422/849	113/231	113/231	0
XPOWELL	1000	568/1175	212/473	97/229	3
TRIGON	1000	231/467	40/92	40/92	0

Tablica(1.1) Iteracije i funkcije/gradijenti procjene zahtijevane od nelinearne metode konjugiranih gradijenata na različite probleme.

Slika (1.7) prikazuje konvergenciju nelinearne metode konjugiranih gradijenata na kompliciranoj funkciji s mnogo lokalnih minimuma i maksimuma. Možemo vidjeti i puteve konvergencije dosadašnjih metoda konjugiranih gradijenata, te usporediti koja je najbolja metoda rješavanja ovog problema.



Slika (1.7)

(a) Komplicirana funkcija s mnogo lokalnih minimuma i maksimuma.

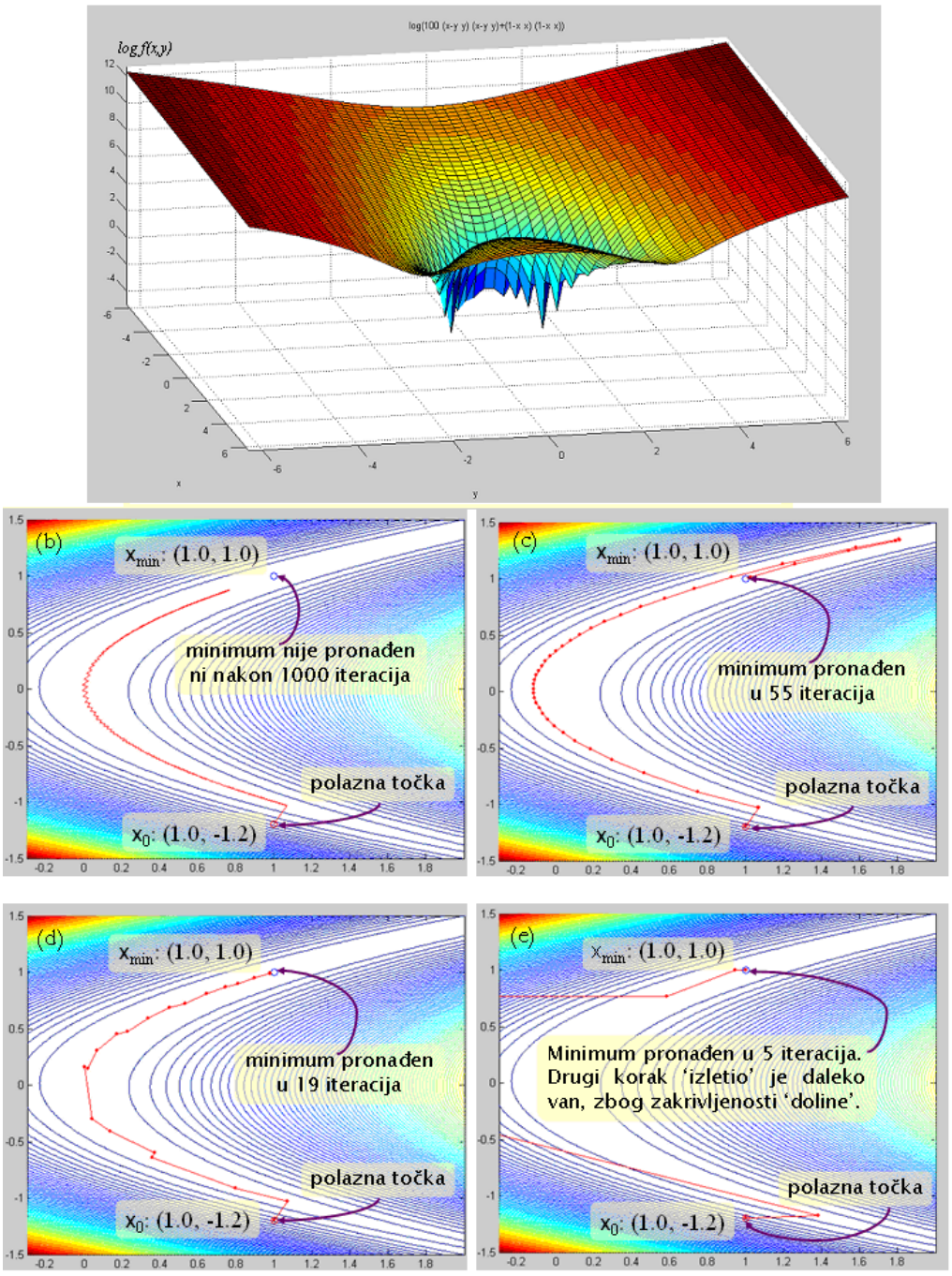
(b) Put konvergencije Fletcher-Reevesove metode KG.

(c) Put konvergencije Polak-Ribiereove metode KG.

(d) Nelinearna metoda KG može biti mnogo učinkovitija s periodičnom reinicijalizacijom.

(e) Prekondicionirana nelinearna metoda KG koja koristi Polak-Ribiereovu formulu i dijagonalnu prekondicionarnu matricu.

Usporedba numeričkih algoritama za optimizaciju nelinearnih funkcija izvodi se na Rosenbrockovoj funkciji koja ima minimum u zakrivljenoj i vrlo sporo spuštajućoj „dolini“.



Slika (1.8) (a) Rosenbrockova funkcija. (b) Metoda najstrmijeg spusta za Rosenbrockovu funkciju. (c) Metoda konjugiranih gradijenata (Fletcher-Reeves) za Rosenbrockovu funkciju. (d) Metoda konjugiranih gradijenata (Polak-Ribiere) za Rosenbrockovu funkciju. (e) Newtonova metoda na Rosenbrockovoj funkciji

BILJEŠKE

Metodu konjugiranih gradijenata su razvili 1950tih Hestnes i Stiefel, kao alternativnu metodu faktorizaciji za nalaženje rješenja simetričnih, pozitivno definitnih sustava. Tek nekoliko godina poslije, u jednom od najvažnijih razvoja u rijetkoj linearnoj algebri, ova je metoda počela biti gledana kao iterativna metoda koja može dati dobru aproksimaciju rješenja u mnogo manje od *n* koraka.
Dovoljno je interesantno da je nelinearna metoda konjugiranih gradijenata Fletchera i Reevesa prestavljena nakon što je linearna metoda konjugiranih gradijenata prestala biti popularna, ali nekoliko godina prije nego je ponovno otkivena kao iterativna metoda za linearne sustave. Hager i Zhang su objavili neke od najboljih računarskih rezultata dobivenih s nelinearnim metodama KG. Njihova implemetacija se bazira na formuli (1.50) i koristi visoko precizne procedure jednodimenzionalne optimizacije. Ovaj rezultat je prikazan u *tablici (1.1)* uzetoj od Gilberta i Nocedala. Oni također opisuju jednodimenzionalnu optimizaciju koja garantira da *algoritam PR+* uvijek generira smjerove silaska i dokazuju globalnu konvergenciju.
Analiza pripisana Powelu omogućuje buduću evidenciju neučinkovitosti FR metode koristeći egzaktne jednodimenzionalne optimizacije. On pokazuje da ako iteracije uđu u područje u kojem je funkcija dvodimenzionalno kvadratna

$$f(x) = \frac{1}{2} x^T x ,$$

onda kut između gradijenta ∇f_k i smjera silaska p_k ostaje konstantan. Kako ovaj kut može biti proizvoljno blizu 90° , FR metoda može biti sporija od metode najbržeg silaska. Polak-Ribiereova metoda se ponaša poprilično drukčije u ovim uvjetima: ako je vrlo mali korak generiran, sljedeći smjer teži smjeru najbržeg silaska, kao što smo se složili gore. Ovo svojstvo omogućuje niz malih koraka.
Većina teorija o brzini konvergencije konjugiranih gradijenata pretpostavlja da je jednodimenzionalna optimizacija egzaktna. Crowder i Wolfe su pokazali da je brzina konvergencije linearna. Powel proučava slučaj u kojem metoda konjugiranih gradijenata ulazi u područje gdje je ciljna funkcija kvadratna i pokazuje da su i zavšavanje u konačno koraka i brzina konvergencije linearni. Cohen i Burmeister su dokazali kvadratnu konvergenciju (1.51) za općenite ciljne funkcije. Ritter pokazuje da je u stvari ta brzina super kvadratna tj.

$$\|x_{k+n} - x^*\| = o(\|x_k - x^*\|^2) .$$

Powel daje za nijansu bolje rješenje i izvodi numeričke testove na malim problemima da izmjeri brzinu primjećenu u praksi. On također sumira brzine kovergencije za asimptotski egzaktne jednodimenzionalne optimizacije. Čak i veće brzine konvergencije mogu biti postignute pod pretpostavkom da su smjerovi silaska uniformno linearno neovisni, ali ova pretpostavka je teška za dokazati i ne pojavljuje se često u praksi.
Nemirowsky i Yudin su posvetili pažnju globalnoj učinkovitosti FR i PR metoda sa egzaktnim jednodimenzionalnima optimizacijama. Pokazali su da za stogo konveksne probleme, ne samo da FR i PR metode ne uspijevaju postići optimalnu granicu, nego da i mogu biti sporije od metode najbržeg silaska.

DODATAK

Na kraju ovog rada se nalazi CD s implementiranim algoritmima (1.2), (1.3), (1.4) i (1.5) u programskom jeziku Dev C/C++. Matrice, vektori i funkcije su redom iz zadanih primjera (1.1), (1.2) i (1.3).

Algoritam(1.2) koristimo da riješimo sustav linearnih jednadžbi $Ax = b$ gdje su elementi matrice A zadani s $A_{i,j} = 1/(i + j + 1)$, te je $b = (1,1,...,1)^T$ i $x_0 = 0$. Prvo učitamo dimenziju matrice A te zatim brojimo iteracije dokle god rezidual ne bude manji od 10^{-6}. Program ispisuje rješenje x i broj iteracija koji je bio potreban da bismo došli do rješenja.

Algoritam(1.3) rješava isti problem uz korišenje prekondicionarne matrice M. Kako M mora biti simetrična i pozitivno definitna matrica, uzeli smo dijagonalnu matricu gdje su elementi na dijagonali zadani s $M_{i,i} = 1 - 1/\sqrt{i+1}$. Prvo učitamo dimenziju matrice A te zatim brojimo iteracije dokle god rezidual ne bude manji od 10^{-6}. Program ispisuje rješenje x i broj iteracija koji je bio potreban da bismo došli do rješenja.

Algoritmi(1.4) i (1.5) prestavljaju Fletcher-Reevesovu i Polak-Ribiereovu metodu konjugiranih gradijenata. Metode ispitujemo na Rosenbrockovoj funkciji $f(x, y) = 100(x - y^2)^2 + (1 - x^2)^2$. Početna točka obje metode je (0.25,0).
Programi ispisuje rješenje (x,y) i broj iteracija koji je bio potreban da bismo došli do rješenja. Rješenja obje metode (x,y) smo spremili u datoteke rosen1.txt i rosen2.txt te ih nacrtali u MATLABu.
Kod za ove dvije izvedbe u MATLABu se nalazi na slikama matlab1.bmp i matlab2.bmp te slike samih izvedbi u datotekama matlabFR.bmp i matlabPR.bmp.

LITERATURA

JORGE NOCEDAL AND STEPHEN J. WRIGHT, ***Numerical optimization (2nd ed) (Springer, 2006)***

JONATHAN RICHARD SHEWCHUK, ***An introduction to the conjugate gradient method without agonizing pain (Carnegie Mellon University, Pittsburgh, PA 15213)***

R. FLETCHER AND C. M. REEVES, *Function minimization by conjugate gradients*, Computer Journal, 7 (1964), pp. 149–154.

E. POLAK AND G. RIBIÈRE, *Note sur la convergence de méthodes de directions conjuguées*, Revue Française d'Informatique et de Recherche Opérationnelle, 16 (1969), pp. 35–43.

M. J. D. POWELL, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Computer Journal, 91 (1964), pp. 155–162.

———, *Restart procedures for the conjugate gradient method*, Mathematical Programming, 12 (1977), pp. 241–254.

———, *Nonconvex minimization calculations and the conjugate gradient method*, Lecture Notes in Mathematics, 1066 (1984), pp. 122–141.

———, *Some convergence properties of the conjugate gradient method*, Mathematical Programming, 11 (1976), pp. 42–49.

D. LUENBERGER, *Introduction to Linear and Nonlinear Programming*, Addison Wesley, second ed., 1984.

W. W. HAGER AND H. ZHANG, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM Journal on Optimization, 16 (2005), pp. 170–192.

J. GILBERT AND J. NOCEDAL, *Global convergence properties of conjugate gradient methods for optimization*, SIAM Journal on Optimization, 2 (1992), pp. 21–42.

M. AL-BAALI, *Descent property and global convergence of the Fletcher-Reeves method with inexact line search*, I.M.A. Journal on Numerical Analysis, 5 (1985), pp. 121–124.

Y. DAI AND Y. YUAN, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM Journal on Optimization, 10 (1999), pp. 177–182.

J. GILBERT AND J. NOCEDAL, *Global convergence properties of conjugate gradient methods for optimization*, SIAM Journal on Optimization, 2 (1992), pp. 21–42.

W. W. HAGER AND H. ZHANG, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM Journal on Optimization, 16 (2005), pp. 170–192.

———, *A survey of nonlinear conjugate gradient methods*. To appear in the Pacific Journal of Optimization, 2005.

A. COHEN, *Rate of convergence of several conjugate gradient algorithms*, SIAM Journal on Numerical Analysis, 9 (1972), pp. 248–259.

W. BURMEISTER, *Die konvergenzordnung des Fletcher-Powell algorithmus*, Zeitschrift für Angewandte Mathematik und Mechanik, 53 (1973), pp. 693–699.

H. P. CROWDER AND P. WOLFE, *Linear convergence of the conjugate gradient method*, IBM Journal of Research and Development, 16 (1972), pp. 431–433.

A. S. NEMIROVSKII AND D. B. YUDIN, *Problem complexity and method efficiency*, John Wiley & Sons, New York, 1983.

K. RITTER, *On the rate of superlinear convergence of a class of variable metric methods*, Numerische Mathematik, 35 (1980), pp. 293–313.