

# Globalno konvergentne modifikacije Newtonove metode

---

Kuhar, Lara

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:263486>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-27**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



**SVEUČILIŠTE U ZAGREBU**  
**PRIRODOSLOVNO–MATEMATIČKI FAKULTET**  
**MATEMATIČKI ODSJEK**

Lara Kuhar

**GLOBALNO KONVERGENTNE**  
**MODIFIKACIJE NEWTONOVE**  
**METODE**

Diplomski rad

Voditelj rada:  
doc. dr. sc. Tina Bosner

Zagreb, 2021.

Ovaj diplomski rad obranjen je dana \_\_\_\_\_ pred ispitnim povjerenstvom u sastavu:

1. \_\_\_\_\_, predsjednik
2. \_\_\_\_\_, član
3. \_\_\_\_\_, član

Povjerenstvo je rad ocijenilo ocjenom \_\_\_\_\_.

Potpisi članova povjerenstva:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

# Sadržaj

<b>Sadržaj</b>	<b>iii</b>
<b>Uvod</b>	<b>1</b>
<b>1 Newtonova metoda za rješavanje nelinearnih jednadžbi i bezuvjetne minimizacije</b>	<b>2</b>
1.0.1 Konvergencija . . . . .	3
1.0.2 Greške zaokruživanja i konačna aritmetika . . . . .	7
1.1 Newtonova metoda za rješavanje sistema nelinearnih jednadžbi . . . . .	8
1.2 Lokalna konvergencija Newtonove metode . . . . .	10
1.3 Newtonova metoda bezuvjetne minimizacije . . . . .	11
<b>2 Uvod u globalne metode za nelinearne jednadžbe i bezuvjetnu minimizaciju</b>	<b>15</b>
2.1 Kvazi-Newtonov okvir . . . . .	15
2.2 Smjer silaska . . . . .	16
<b>3 Line search - pretraživanje linija</b>	<b>19</b>
3.1 Wolfe-ovi uvjeti i konvergentni rezultati . . . . .	21
3.1.1 Wolfe-ovi uvjeti . . . . .	21
3.1.2 Konvergencija . . . . .	23
3.2 Odabir koraka pomoću backtrackinga . . . . .	29
<b>4 Trust region model - Model regije povjerenja</b>	<b>32</b>
4.1 Lokalno ograničen optimalni (“hook”) korak . . . . .	35
4.2 Dvostruki dogleg korak . . . . .	39
4.3 Modifikacija trust region algoritma . . . . .	43
<b>5 Globalne metode za rješavanje sustava nelinearnih jednadžbi</b>	<b>47</b>
<b>6 Zaustavljanje, skaliranje i testiranje</b>	<b>52</b>
6.1 Skaliranje . . . . .	52

<i>SADRŽAJ</i>	iv
6.2 Kriterij zaustavljanja . . . . .	54
6.3 Testiranje . . . . .	56
<b>Bibliografija</b>	<b>60</b>

# Uvod

U primjeni se često pojavljuje potreba za rješavanjem bezuvjetnih minimizacijskih problema ili rješavanjem sistema nelinearnih jednadžbi. Za oba problema koriste se skoro iste numeričke metode, gdje je jedna od najpopularnijih Newtonova metoda. Poznat je kao jedan od glavnih nedostataka Newtonove metode nepostojanje globalne konvergencije, pa je cilj ovog diplomskog rada ponuditi nekoliko mogućih modifikacija Newtonove metode kojima se taj problem rješava.

Na početku se definira Newtonova metoda za jednovarijabilne nelinearne jednadžbe i jednodimenzionalne bezuvjetne minimizacije te se uvode potrebne definicije i teoremi za daljnje razumijevanje rada algoritama. Promatra se primjena Newtonove metode na viševarijabilnim i višedimenzionalnim problemima, njena lokalna konvergencija i njeni nedostaci. Naposljetku se uvode globalne metode - line search i trust region algoritmi, koji u kombinaciji s Newtonovom metodom osiguravaju globalnu i lokalno brzu konvergenciju.

# Poglavlje 1

## Newtonova metoda za rješavanje nelinearnih jednažbi i bezuvjetne minimizacije

Ideja Newtonove metode, poznate i kao metoda tangente, za rješavanje nelinearnih problema jedne varijable,

za danu  $f : \mathbb{R} \rightarrow \mathbb{R}$  nađi  $x_*$  takav da vrijedi  $f(x_*) = 0$ ,

je da se graf funkcije  $f$  aproksimira jednažbom tangente za danu točku  $x_c$ . Newtonova metoda u svakoj iteraciji aproksimira funkciju  $f$  lokalnim afnim modelom

$$M_c(x) = f(x_c) + f'(x_c)(x - x_c) \quad (1.1)$$

i traži njegov korijen, koji je tada dan sa

$$x_+ = x_c - \frac{f(x_c)}{f'(x_c)}. \quad (1.2)$$

Do Newtonove metode se može doći i razvojem funkcije  $f$  u Taylorov red oko točke  $x_c$

$$f(x) = f(x_c) + f'(x_c)(x - x_c) + \frac{f''(x_c)(x - x_c)^2}{2!} + \dots = \sum_{i=0}^{\infty} \frac{f^{(i)}(x_c)(x - x_c)^i}{i!}$$

i aproksimacijom  $f(x)$  afnim dijelom Taylorovog reda, koji je prirodno dan sa (1.1), pri čemu je korijen opet dan sa (1.2). Međutim, Dennis i Schnabel [1] preferiraju drugačiji pristup. Umjesto raznih pretpostavaka o derivacijama višeg reda, koje postaju kompliciranije, Newtonova metoda se jednostavno i prirodno izvede iz Newtonog teorema

$$f(x) = f(x_c) + \int_{x_c}^x f'(z) dz,$$

gdje se čini razumnim određeni integral aproksimirati sa

$$\int_{x_c}^x f'(z)dz = f'(x_c)(x - x_c),$$

i zadržati afinu aproksimaciju od  $f(x_c)$  danu sa (1.1).

Slično, kod problema bezuvjetne minimizacije jedne varijable, Newtonov korak za traženje rješenja  $f'(x) = 0$  uz danu početnu točku  $x_c$ , dan je sa

$$x_+ = x_c - \frac{f'(x_c)}{f''(x_c)}. \quad (1.3)$$

Newtonov korak je dobiven primjenom afinog modela od  $f'(x)$  oko točke  $x_c$ , što je ekvivalentno modelu kvadratnog polinoma od  $f(x)$  oko  $x_c$ ,

$$m_c(x) = f(x_c) + f'(x_c)(x - x_c) + \frac{1}{2}f''(x_c)(x - x_c)^2.$$

### 1.0.1 Konvergencija

Obzirom da je Newtonova metoda iterativna metoda, zanimat će nas konvergira li niz točaka  $x_1, x_2, \dots$ , krećući od početne točke  $x_0$ , prema rješenju  $x_*$  i ako konvergira, koliko je brza konvergencija. Definirajmo najprije pojam konvergencije niza, a zatim i tipove konvergencija te pomoćne iskaze koji će se primijeniti u nastavku rada.

**Definicija 1.0.1.** Niz realnih brojeva  $(a_n)_n$  konvergira ili teži k realnom broju  $a \in \mathbb{R}$  ako svaki otvoreni interval polumjera  $\varepsilon$  oko točke  $a$  sadrži gotovo sve članove niza, tj.

$$(\forall \varepsilon > 0), (\exists n_\varepsilon \in \mathbb{N}), (\forall n \in \mathbb{N}), ((n > n_\varepsilon) \Rightarrow (|a_n - a| < \varepsilon)).$$

Tada  $a$  zovemo granična vrijednost ili limes niza  $(a_n)_n$  i pišemo  $a = \lim_{n \rightarrow \infty} a_n$ . Ako niz ne konvergira, onda kažemo da on divergira.

**Definicija 1.0.2.** Neka je  $x_* \in \mathbb{R}$ ,  $x_k \in \mathbb{R}$ ,  $k = 0, 1, 2, \dots$ . Tada niz  $\{x_k\} = \{x_0, x_1, x_2, \dots\}$  konvergira prema  $x_*$  ako vrijedi

$$\lim_{k \rightarrow \infty} |x_k - x_*| = 0.$$

Ako dodatno, postoji konstanta  $c \in [0, 1)$  i  $\hat{k} \geq 0$  takav da za sve  $k \geq \hat{k}$  vrijedi

$$|x_{k+1} - x_*| \leq c|x_k - x_*|, \quad (1.4)$$

kažemo da  $\{x_k\}$  konvergira  $q$ -linearno prema  $x_*$ . Ako za neki niz  $\{c_k\}$  koji konvergira prema 0, vrijedi

$$|x_{k+1} - x_*| \leq c_k|x_k - x_*|, \quad (1.5)$$



kažemo da  $\{x_k\}$  konvergira  $q$ -superlinearno prema  $x_*$ . Ako postoji konstanta  $p > 1$ ,  $c \geq 0$  i  $\hat{k} \leq 0$  takav da  $\{x_k\}$  konvergira prema  $x_*$  i da za sve  $k \geq \hat{k}$  vrijedi

$$|x_{k+1} - x_*| \leq c|x_k - x_*|^p, \quad (1.6)$$

kažemo da  $\{x_k\}$  konvergira prema  $x_*$   $q$ -redom najmanje  $p$ . Ako je  $p = 2$  ili  $p = 3$ , kažemo da je konvergencija  $q$ -kvadratna ili  $q$ -kubična, respektivno.

Nadalje, ako  $\{x_k\}$  konvergira prema  $x_*$  i umjesto (1.5), vrijedi

$$|x_{k+j} - x_*| \leq c_k|x_k - x_*|$$

za neki fiksni  $j$ , kažemo da je  $\{x_k\}$   $j$ -korak  $q$ -superlinearno konvergentan prema  $x_*$ . Ako  $\{x_k\}$  konvergira prema  $x_*$  i umjesto (1.6), za  $k \geq \hat{k}$  vrijedi

$$|x_{k+j} - x_*| \leq c|x_k - x_*|^p$$

za neki fiksni  $j$ , kažemo da je  $\{x_k\}$   $j$ -korak  $q$ -reda (reda barem  $p$ ) konvergentan.

U praksi  $q$ -linearna konvergencija zna biti vrlo spora, dok su  $q$ -kvadratna i  $q$ -kubična vrlo brze konvergencije. Međutim njihovo ponašanje ovisi o konstanti  $c$  u (1.4) - (1.6). Manji  $c$  – bolja konvergencija, veći  $c$  – slabija konvergencija.  $q$ -superlinearna konvergencija je direktno povezana s brojem potrebnih iteracija za smanjenje  $c_k$ , pri čemu  $q$  označava “kvocijent”. Za iterativnu metodu koja, počevši dovoljno blizu točnog rješenja, konvergira prema tom rješenju kažemo da je *lokalno konvergentna*.

Definirajmo sada pojam Lipschitzove neprekidnosti.

**Definicija 1.0.3.** Funkcija  $g$  je Lipschitz neprekidna s konstantom  $\gamma$  na skupu  $X$  i pišemo  $g \in Lip_\gamma(X)$ , ako za svaki  $x, y \in X$  vrijedi

$$|g(x) - g(y)| \leq \gamma|x - y|.$$

Sljedeća lema ukazuje da ako je  $f'(x)$  Lipschitz neprekidna, tada možemo dobiti granice koliko je afina aproksimacija  $f(x) + f'(x)(y - x)$  blizu  $f(y)$ .

**Lema 1.0.4.** Za otvoreni interval  $D$ , neka je  $f : D \rightarrow \mathbb{R}$  i  $f' \in Lip_\gamma(D)$ . Tada je za bilo koji  $x, y \in D$

$$|f(y) - f(x) - f'(x)(y - x)| \leq \frac{\gamma(y - x)^2}{2}. \quad (1.7)$$

Prednost Lipschitzove neprekidnosti je u tome što nije potrebno računati sljedeću veću derivaciju, što je vrlo korisno u višedimenzionalnim problemima. Idući teorem je osnovni teorem numeričke matematike.

**Teorem 1.0.5.** *Neka je  $f : D \rightarrow \mathbb{R}$  za otvoreni interval  $D$  i neka je  $f' \in \text{Lip}_\gamma(D)$ . Pretpostavimo da je za neki  $\rho > 0$ ,  $|f'(x)| \geq \rho$  za svaki  $x \in D$ . Ako  $f(x) = 0$  ima rješenje  $x_* \in D$ , tada postoji  $\eta > 0$  takav da: ako je  $|x_0 - x_*| < \eta$ , tada postoji niz  $\{x_k\}$  generiran sa*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots$$

*i konvergira prema  $x_*$ . Nadalje, za  $k = 0, 1, \dots$*

$$|x_{k+1} - x_*| \leq \frac{\gamma}{2\rho} |x_k - x_*|^2. \quad (1.8)$$

Nadalje u ovom poglavlju razmatramo lokalne algoritme sistema nelinearnih jednadžbi i bezuvjetne minimizacije za probleme s  $n \geq 2$  varijabli. Izvodimo Newtonovu metodu za rješavanje sistema nelinearnih jednadžbi, pokazujemo da Newtonova metoda za većinu problema konvergira  $q$ -kvadratno, ali ne postiže nužno i globalnu konvergenciju, te uvodimo verziju Newtonove metode za  $n$ -dimenzionalne bezuvjetne minimizacijske probleme.

Prije samog razmatranja Newtonove metode na viševarijabilnim i višedimenzionalnim problemima, preostaje definirati i iskazati neke bitne pojmove za lakše razumijevanje sadržaja.

**Definicija 1.0.6.** *Kažemo da je neprekidna funkcija  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  neprekidno diferencijabilna u točki  $x \in \mathbb{R}^n$ , ako postoji neprekidna parcijalna derivacija  $\frac{\partial f}{\partial x_i}(x)$  za  $i = 1, \dots, n$ . Tada je gradijent funkcije definiran sa*

$$\nabla f(x) = \left[ \frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right]^T \quad (1.9)$$

*Funkcija  $f$  je neprekidno diferencijabilna na otvorenom skupu  $D \subset \mathbb{R}^n$ , u oznaci  $f \in C^1(D)$ , ako je neprekidno diferencijabilna u svakoj točki od  $D$ .*

**Lema 1.0.7.** *Neka je  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  neprekidno diferencijabilna na otvorenom konveksnom skupu  $D \subset \mathbb{R}^n$ . Tada za  $x \in D$  i za bilo koju nenul perturbaciju  $p \in \mathbb{R}^n$ , derivacija funkcije  $f$  u točki  $x$  duž vektora  $p$ , definirana sa*

$$\frac{\partial f}{\partial p} = \lim_{\varepsilon \rightarrow 0} \frac{f(x + \varepsilon p) - f(x)}{\varepsilon},$$

*postoji i jednaka je  $\nabla f(x)^T p$ . Za bilo koji  $x, x + p \in D$  vrijedi*

$$f(x + p) = f(x) + \int_0^1 \nabla f(x + tp)^T p \, dt = f(x) + \int_x^{x+p} \nabla f(z) dz, \quad (1.10)$$

*i postoji  $z \in (x, x + p)$  takav da je*

$$f(x + p) = f(x) + \nabla f(z)^T p. \quad (1.11)$$

**Definicija 1.0.8.** *Neprekidno diferencijabilna funkcija  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  je dva puta neprekidno diferencijabilna u  $x \in \mathbb{R}^n$ , ako postoji neprekidna druga parcijalna derivacija  $\frac{\partial^2 f}{\partial x_i \partial x_j}(x)$ . Tada je Hesseova matrica funkcije  $f$  u točki  $x$  definirana kao  $n \times n$  matrica sa  $(i, j)$  elementima*

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}, \quad 1 \leq i, j \leq n. \quad (1.12)$$

*Funkcija  $f$  je dva puta neprekidno diferencijabilna na otvorenom skupu  $D \subset \mathbb{R}^n$ , u oznaci  $f \in C^2(D)$ , ako je dva puta diferencijabilna u svakoj točki u  $D$ .*

**Lema 1.0.9.** *Neka je  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  dva puta neprekidno diferencijabilna na otvorenom konveksnom skupu  $D \subset \mathbb{R}^n$ . Tada za bilo koji  $x \in D$  i neku nenul perturbaciju  $p \in \mathbb{R}^n$ , druga derivacija funkcije  $f$  u točki  $x$  duž smjera  $p$ , definirana sa*

$$\frac{\partial^2 f}{\partial p^2}(x) = \frac{\frac{\partial f}{\partial p}(x + \varepsilon p) - \frac{\partial f}{\partial p}(x)}{\varepsilon},$$

*postoji i jednaka je  $p^T \nabla^2 f(x) p$ . Za bilo koji  $x, x + p \in D$ , postoji  $z \in (x, x + p)$  takav da je*

$$f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(z) p. \quad (1.13)$$

Hesseova matrica je uvijek simetrična sve dok je  $f$  dva puta neprekidno diferencijabilna i upravo gornja lema sugerira modeliranje funkcije  $f$  u točki  $x_c$  modelom kvadratnog polinoma

$$m_c(x_c + p) = f(x_c) + \nabla f(x_c)^T p + \frac{1}{2} p^T \nabla^2 f(x_c) p.$$

**Definicija 1.0.10.** *Neprekidna funkcija  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  je neprekidno diferencijabilna u  $x \in \mathbb{R}^n$ , ako je svaka komponentna funkcija  $f_i$ ,  $i = 1, \dots, m$  neprekidno diferencijabilna u  $x$ . Matrica derivacija od  $F$  u  $x$  se ponekad naziva Jacobijeva matrica od  $F$  u točki  $x$ , a njena transponirana matrica se naziva gradijent funkcije  $F$  u  $x$ . Uobičajene oznake su:*

$$F'(x) \in \mathbb{R}^{m \times n}, \quad F'(x)_{ij} = \frac{\partial f_i}{\partial x_j}(x), \quad F'(x) = J(x) = \nabla F(x)^T.$$

*$F$  je neprekidno diferencijabilna na otvorenom skupu  $D \subset \mathbb{R}^n$ , u oznaci  $F \in C^1(D)$ , ako je  $F$  dva puta neprekidno diferencijabilna u svakoj točki skupa  $D$ .*

**Definicija 1.0.11.** *Neka su  $m, n > 0$ ,  $G : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$ ,  $x \in \mathbb{R}^n$  i neka  $\|\cdot\|$  norma na  $\mathbb{R}^n$ , a  $\|\|\cdot\|\|$  norma na  $\mathbb{R}^{m \times n}$ .  $G$  je Lipschitz neprekidna u  $x$ , ako postoji otvoreni skup  $D \subset \mathbb{R}^n$ ,  $x \in D$  i konstanta  $\gamma$  takva da za sve  $v \in D$  vrijedi*

$$\|\|G(v) - G(x)\|\| \leq \gamma \|v - x\|. \quad (1.14)$$

Konstanta  $\gamma$  naziva se Lipschitzova konstanta od  $G$  u  $x$ . Za bilo koji skup  $D$  koji sadrži  $x$  za koji vrijedi (1.14),  $G$  je Lipschitz neprekidna u  $x$  u okolini od  $D$ . Ako vrijedi (1.14) za svaki  $x \in D$ , tada je  $G \in Lip_\gamma(D)$ .

**Lema 1.0.12.** Neka je  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  neprekidno diferencijabilna na otvorenom skupu  $D \in \mathbb{R}^n$ ,  $x \in D$  i neka je  $J$  Lipschitz neprekidna u  $x$  u okolini skupa  $D$ , koristeći vektorsku normu, matičnu operatorsku normu induciranu danom vektorskom normom i konstantu  $\gamma$ . Tada, za bilo koji  $x + p \in D$ ,

$$\|F(x + p) - F(x) - J(x)p\| \leq \frac{\gamma}{2} \|p\|^2. \quad (1.15)$$

## 1.0.2 Greške zaokruživanja i konačna aritmetika

U praksi se kod računanja javljaju greške zaokruživanja, koje nastaju u računalima, jer koriste konačnu aritmetiku ili binarnu aritmetiku s pomičnom točkom, kod koje je unaprijed rezerviran određen broj binarnih mjesta za eksponent i mantisu. Svaka se računaska operacija dva broja izračunava s nekom malom greškom, koja može biti i nula kada je rezultat egzaktno izračunat. Kada greška nije nula, zovemo je greškom zaokruživanja i možemo je precizno ocijeniti.

Greške zaokruživanja su neizbježne pri svakom zahtjevnijem računanju, jer računala koriste konačnu aritmetiku. Već i samo uskladištavanje podatka u računalo dovodi do grešaka jer se svaki broj mora reprezentirati konačnim brojem binarnih znamenaka. Brojevi smješteni u računalo, uglavnom nisu njihove točne, nego približne binarne reprezentacije, koje se obično označavaju sa  $fl(x)$  i nazivaju *floating point* reprezentacija.

Pretpostavi li se da su  $x$  i  $y$  brojevi smješteni u računalo,  $x = fl(x)$  i  $y = fl(y)$ , vrijedi

$$fl(x \circ y) = (x \circ y)(1 + \varepsilon), \quad |\varepsilon| \leq u, \quad \circ \in \{+, -, *, /\},$$

pri čemu su  $+, -, *, /$  operacije u računalu,  $\varepsilon$  greška zaokruživanja i  $u$  je tzv. *preciznost računanja* ili *strojni  $u$* .  $u$  je uniformna gornja ograda za sve greške  $\varepsilon$  na danom računalu i za računala s binarnom aritmetikom sa  $p$  binarnih znamenaka u mantisi, vrijedi  $u = 2^{-p+1}$  ili  $u = 2^{-p}$ , ovisno o načinu zaokruživanja koje računalo koristi. Razlikujemo apsolutnu,  $|x - y|$ , i relativnu,  $\frac{|x-y|}{|x|}$ , grešku zaokruživanja. U ovom radu se uglavnom primjenjuje relativna greška, eventualno u slučaju kada je  $x = 0$ , tada se primjenjuje apsolutna.

Određeni aspekti algoritama, poput kriterija zaustavljanja, ovise o strojnoj preciznosti, koju je stoga potrebno karakterizirati kako bi sva razmatranja i računalni programi bili neovisni o određenom stroju. Najčešća je primjena takozvanog *machine epsilon* (strojnog  $\varepsilon$ ), definiranog kao najmanji pozitivni broj  $\tau$  takav da je  $1 + \tau > 1$  na danom računalu. U daljnjem tekstu označava se sa *macheps*, a algoritam izračuna je sljedeći:

**Algoritam 1.0.13. Algoritam za izračun machine epsilon-a**

Postavi  $macheps = 1$

Sve dok je  $(1 + macheps) = 1$ , radi sljedeće:

$macheps = macheps/2$

$macheps = 2macheps$ .

## 1.1 Newtonova metoda za rješavanje sistema nelinearnih jednadžbi

Osnovni problem proučavanja je rješavanje problema sistema nelinearnih jednadžbi:

$$\text{Za danu funkciju } F : \mathbb{R}^n \longrightarrow \mathbb{R}^n, \quad \text{nađi } x_* \in \mathbb{R}^n \text{ takav da je } F(x_*) = 0, \quad (1.16)$$

uz pretpostavku da je  $F$  neprekidno diferencijabilna. Uvodimo Newtonovu metodu za dani problem, kao i motivacijske faktore za daljnje modifikacije Newtonove metode te kreiranje globalno jakih i lokalno brzih algoritama. Pritom se ignorira postojanje i jedinstvenost rješenja problema (1.16) i pretpostavlja se da u praksi to neće biti problem.

Newtonova metoda za problem (1.16) temelji se na traženju korijena affine aproksimacije funkcije  $F$  u trenutnoj točki iteracije  $x_c$ . Aproksimacija funkcije  $F$  se kreira kao i u slučaju problema jedne varijable. Integral izraza

$$F(x_c + p) = F(x_c) + \int_{x_c}^{x_c+p} J(z) dz \quad (1.17)$$

aproksimira se linearnim izrazom  $J(x_c) \cdot p$  kako bi se dobila aproksimacija od  $F$  u točki  $x_c$  sa perturbacijom  $p$ ,

$$M_c(x_c + p) = F(x_c) + J(x_c) \cdot p. \quad (1.18)$$

Zatim se za korak  $s^N$ , za koji je  $M_c(x_c + s^N) = 0$ , rješava Newtonova iteracija problema (1.16),

$$\begin{aligned} J(x_c)s^N &= -F(x_c), \\ x_+ &= x_c + s^N \end{aligned} \quad (1.19)$$

Ekvivalentan postupak je traženje korijena afinog modela funkcije  $F$  s  $n$  komponenata danog sa

$$(M_c)_i(x_c + s^N) = f_i(x_c) + \nabla f_i(x_c)^T s^N, \quad i = 1, \dots, n.$$

**Algoritam 1.1.1. Newtonova metoda za sustave nelinearnih jednadžbi**

Za neprekidnu diferencijabilnu funkciju  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  i  $x_0 \in \mathbb{R}^n$ : kod svake iteracije  $k$ , riješi:

$$J(x_k)s_k = -F(x_k),$$

$$x_{k+1} = x_k + s_k.$$

Prednost Newtonove metode je u tome da, ako je početna točka  $x_0$  dovoljno blizu rješenja  $x_*$  i ako je  $J(x_*)$  regularna matrica, niz točaka  $x_k$ , generiran Algoritmom 1.1.1 konvergira  $q$ -kvadratno prema  $x_*$ . Također, Newtonova metoda je dobra za rješavanje gotovo linearnih problema, jer u jednoj iteraciji pronalazi korijen afine funkcije te ako je neka komponenta funkcije  $F$  afina, svaka iteracija Newtonove metode će biti rješenje tih jednadžbi. Na primjer, ukoliko je  $f_1$  afina funkcija, tada je

$$f_1(x_1) = f_1(x_2) = f_1(x_3) = \dots = 0.$$

S druge strane, Newtonova metoda nije previše uspješna ukoliko je vrlo loše procijenjena početna točka  $n$ -varijabilnog nelinearnog problema. Prema tome, karakteristika konvergencije Newtonove metode određuje primijenu  $n$ -dimenzionalnih algoritama, pri čemu se Newtonova metoda, zbog brze lokalne konvergencije, uvijek želi iskoristiti, barem u posljednjim iteracijama nekog nelinearnog algoritma, ali ju je poželjno modificirati zbog globalne konvergencije.

Problem implementacije iteracija Algoritma 1.1.1 je što  $J(x_k)$  može biti singularna ili loše uvjetovana, zbog čega se linearni sustav  $J(x_k)s_k = -F(x_k)$  ne može pouzdano riješiti za korak  $s_k$ . Moguća modifikacija Algoritma 1.1.1 je dovoljna pertubacija od  $J(x_k)$ , da postane dobro uvjetovana<sup>1</sup> matrica i da algoritam nastavi s iteracijama. Međutim, ta modifikacija se ne preporuča, već se u slučaju loše uvjetovanosti direktno nastavlja globalnom metodom nelinearnih jednadžbi (Poglavlje 5) koje odgovaraju perturbiranom linearnom modelu (1.18).

Spomenute prednosti i mane Newtonove metode, ključ su razvoja višedimenzionalnih algoritama, obzirom da ističu svojstva koja se žele zadržati i područja koje je potrebno optimizirati ili modificirati.

---

<sup>1</sup>Uvjetovanost matrice,  $cond A = \|A^{-1}\| \|A\| \geq 1$ , se koristi kao pokazatelj osjetljivosti rješenja problema na ulazne podatke. Matrica je singularna kada je  $cond A = \infty$ .

## 1.2 Lokalna konvergencija Newtonove metode

Konvergencija Newtonove metode je od velikog značaja za daljnji razvoj globalnih metoda i većinu dokaza konvergencije u nastavku ovog rada, stoga je potrebno dokazati njenu  $q$ -kvadratnu konvergenciju. Za danu normu  $\|\cdot\|$ , definira se otvorena okolina oko točke  $x$  radijusa  $r$ ,  $N(x, r) = \{\bar{x} \in \mathbb{R}^n : \|\bar{x} - x\| < r\}$ .

**Teorem 1.2.1.** *Neka je  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  neprekidno diferencijabilna na otvorenom konveksnom skupu  $D \subset \mathbb{R}^n$ . Pretpostavimo da postoji  $x_* \in \mathbb{R}^n$  i  $r, \beta > 0$ , takvi da je  $N(x_*, r) \subset D$ ,  $F(x_*) = 0$ ,  $J(x_*)^{-1}$  postoji uz uvjet  $\|J(x_*)^{-1}\| \leq \beta$  i  $J \in \text{Lip}_\gamma(N(x_*, r))$ . Tada postoji  $\varepsilon > 0$  takav da je za sve  $x_0 \in N(x_*, \varepsilon)$  niz  $x_1, x_2, \dots$  generiran sa*

$$x_{k+1} = x_k - J(x_k)^{-1}F(x_k), \quad k = 0, 1, \dots$$

dobro definiran i konvergira prema  $x_*$  zadovoljavajući

$$\|x_{k+1} - x_*\| \leq \beta\gamma\|x_k - x_*\|^2, \quad k = 0, 1, \dots \quad (1.20)$$

*Dokaz.* Odaberemo  $\varepsilon$  tako da je  $J(x)$  regularna za svaki  $x \in N(x_*, \varepsilon)$  i tada pokazujemo da, obzirom da je lokalna greška afinog modela korištena za izračun pojedinog koraka iteracije Newtonove metode najviše  $O(\|x_k - x_*\|^2)$ , konvergencija je  $q$ -kvadratna. Neka je

$$\varepsilon = \min \left\{ r, \frac{1}{2\beta\gamma} \right\}. \quad (1.21)$$

Indukcijom po  $k$  dokazujemo da je u svakom koraku zadovoljeno (1.20) i da vrijedi

$$\|x_{k+1} - x_*\| \leq \frac{1}{2}\|x_k - x_*\|$$

i da je

$$x_{k+1} \in N(x_*, \varepsilon). \quad (1.22)$$

Najprije dokazujemo da je matrica  $J(x_0)$  regularna. Iz  $\|x_0 - x_*\| \leq \varepsilon$ , Lipschitz neprekidnosti od  $J$  u  $x_*$  i (1.21) slijedi

$$\begin{aligned} \|J(x_*)^{-1}[J(x_0) - J(x_*)]\| &\leq \|J(x_*)^{-1}\| \|J(x_0) - J(x_*)\| \\ &\leq \beta\gamma\|x_0 - x_*\| \leq \beta \cdot \gamma \cdot \varepsilon \leq \frac{1}{2}. \end{aligned}$$

Sada prema relaciji <sup>2</sup> slijedi da je  $J(x_0)$  regularna i vrijedi

$$\begin{aligned} \|J(x_0)^{-1}\| &\leq \frac{\|J(x_*)^{-1}\|}{1 - \|J(x_*)^{-1}[J(x_0) - J(x_*)]\|} \\ &\leq 2\|J(x_*)^{-1}\| \\ &\leq 2 \cdot \beta \end{aligned} \quad (1.23)$$

<sup>2</sup>Ako je  $A$  regularna matrica i  $\|A^{-1}(B - A)\| < 1$ , tada je  $B$  regularna i vrijedi  $\|B^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}(B - A)\|}$ , za neku normu  $\|\cdot\|$  na  $\mathbb{R}^n$ . (Dennis, Schnabel [1], Teorem 3.1.4)

Prema tome,  $x_1$  je dobro definiran i vrijedi

$$\begin{aligned} x_1 - x_* &= x_0 - x_* - J(x_0)^{-1}F(x_0) \\ &= x_0 - x_* - J(x_0)^{-1}[F(x_0) - F(x_*)] \\ &= J(x_0)^{-1}[F(x_*) - F(x_0) - J(x_0)(x_* - x_0)]. \end{aligned}$$

Primijetite da je izraz u zagradama samo razlika između  $F(x_*)$  i afinog modela  $M_c(x)$  izvrednjenog u  $x_*$ . Prema tome, temeljem Leme 1.0.12 i (1.23), slijedi

$$\begin{aligned} \|x_1 - x_*\| &\leq \|J(x_0)^{-1}\| \|F(x_*) - F(x_0) - J(x_0)(x_* - x_0)\| \\ &\leq 2\beta \cdot \frac{\gamma}{2} \|x_0 - x_*\|^2 \\ &= \beta\gamma \|x_0 - x_*\|^2, \end{aligned}$$

što dokazuje (1.20). Kako je  $\|x_0 - x_*\| \leq \frac{1}{2\beta\gamma}$ ,

$$\|x_1 - x_*\| \leq \frac{1}{2} \|x_0 - x_*\|,$$

što dokazuje (1.22) i završava slučaj za  $k = 0$ . Na isti način se dokazuje i korak indukcije.  $\square$

Preostaje pitanje granice područja konvergencije Newtonove metode dane Teoremom 1.2.1 u praksi. Granica  $\varepsilon$  u Teoremu 1.2.1 je procjena u najgorem slučaju koja prilično dobro prikazuje koliko se daleko proteže područje kvadratne konvergencije u smjeru od  $x_*$  u kojem je  $F$  najviše nelinearna. S druge strane, u smjeru od  $x_*$  u kojem je  $F$  manje nelinearna, područje konvergencije Newtonove metode može biti puno veće.

### 1.3 Newtonova metoda bezuvjetne minimizacije

Promotrimo sada Newtonovu metodu za problem bezuvjetne minimizacije

$$\min_{x \in \mathbb{R}^n} f : \mathbb{R}^n \longrightarrow \mathbb{R}, \quad (1.24)$$

gdje se pretpostavlja da je  $f$  dva puta neprekidno diferencijabilna. Kao i kod jednodimenzionalne minimizacije, funkcija  $f$  se u trenutnoj točki  $x_c$  modelira kvadratnim polinomom

$$m_c(x_c + p) = f(x_c) + \nabla f(x_c)^T p + \frac{1}{2} p^T \nabla^2 f(x_c) p$$

i rješava se za točku  $x_+ = x_c + s^N$  uz uvjet  $\nabla m_c(x_+) = 0$ , koji je nužan kako bi  $x_+$  bio minimum funkcije  $m_c$ .



**Algoritam 1.3.1. Newtonova metoda za problem bezuvjetne minimizacije**

Za dva puta neprekidnu diferencijabilnu funkciju  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  i  $x_0 \in \mathbb{R}^n$ : kod svake iteracije  $k$ , riješi:

$$\begin{aligned}\nabla^2 f(x_k) s_k^N &= -\nabla f(x_k), \\ x_{k+1} &= x_k + s_k^N.\end{aligned}$$

**Primjer 1.3.2.** Neka je  $f(x_1, x_2) = (x_1 - 2)^4 + (x_1 - 2)^2 x_2^2 + (x_2 + 1)^2$  s minimumom u  $x_* = (2, -1)^T$ . Algoritam 1.3.1, krećući iz točke  $x_0 = (1, 1)^T$ , daje sljedeće iteracije:

$k$	$x_k^T$	$f(x_k)$
0	( 1,0000000 , 1,0000000 )	6,00E+00
1	( 1,0000000 , -0,5000000 )	1,50E+00
2	( 1,3913043 , -0,6956522 )	4,09E-01
3	( 1,7459441 , -0,9487981 )	6,49E-02
4	( 1,9862783 , -1,0482081 )	2,53E-03
5	( 1,9987342 , -1,0001700 )	1,63E-06
6	( 1,9999996 , -1,0000016 )	2,75E-12

Slika 1.1: Tablica iteracija

Algoritam 1.3.1 je jednostavna aplikacija Newtonove metode na sustav  $\nabla f(x) = 0$   $n$ -nelinearnih jednadžbi s  $n$  nepoznanica, jer u svakom koraku iteracije traži korijen afinog modela od  $\nabla f$  definiranog sa  $M_k(x_k + p) = \nabla[m_k(x_k + p)] = \nabla f(x_k) + \nabla^2 f(x_k)p$ . Prednost algoritma je u tom da ako je  $x_0$  dovoljno blizu lokalnog minimuma  $x_*$  od  $f$  i  $\nabla^2 f(x_*)$  regularna (pa i pozitivno definitna prema korolaru <sup>3</sup>), tada će niz  $\{x_k\}$  generiran gornjim algoritmom konvergirati  $q$ -kvadratno prema  $x_*$ . Također, ako je  $f$  striktno konveksna kvadratna funkcija, tada je  $\nabla f$  afina i  $x_1$  će biti jedinstveni minimum.

Nedostatak metode je što nije globalno konvergentna, zahijeva rješenje sustava linearnih jednadžbi i zahtijeva izračune  $\nabla f$  i  $\nabla^2 f$ . Potrebno je uzeti u obzir i činjenicu da, kao lokalna metoda, Newtonova metoda nije sklona minimizacijskom problemu, odnosno ništa u Algoritmu 1.3.1 ne sprječava da iteracije napreduju prema maksimumu ili sedlastoj točki od  $f$ , za koje je  $\nabla f$  također jednak 0. Dakle, svaki korak iteracije u Algoritmu 1.3.1 teži

<sup>3</sup>Neka je  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  dva puta neprekidno diferencijabilna na otvorenom skupu  $D \subset \mathbb{R}^n$ . Ako je  $x_* \in D$  i  $\nabla f(x_*) = 0$  i ako je  $\nabla^2 f$  Lipschitz neprekidna u točki  $x_*$  za regularnu matricu  $\nabla^2 f(x_*)$ , tada je  $x_*$  lokalni minimum od  $f$  ako i samo ako je  $\nabla^2 f(x_*)$  pozitivno definitna. (Dennis, Schanbel [1], Korolar 4.3.3)

kritičnoj točki trenutnog modela lokalnog kvadratnog polinoma, neovisno o tome je li riječ o minimumu, maksimumu ili sedlastoj točki. Konzistentnost algoritma prema lokalnom minimumu funkcije  $f(x)$  javlja se samo kada je  $\nabla^2 f(x_c)$  pozitivno definitna matrica.

Iako se Newtonova metoda primarno koristi kao lokalna metoda kada je  $x_c$  dovoljno blizu minimuma i kada je  $\nabla^2 f(x_c)$  pozitivno definitna, želimo izvesti korisne adaptacije kada  $\nabla^2 f(x_c)$  nije pozitivno definitna. U tom slučaju, Dennis i Schnabel [1] predlažu zamjenu modela Hesseove matrice sa  $\nabla^2 f(x_c) + \mu_c I$ , gdje  $\mu_c > 0$  nije puno veći od najmanjeg  $\mu$  koji matricu  $\nabla^2 f(x_c) + \mu I$  čini pozitivno definitnom i dobro uvjetovanom. Jedan od algoritama za računanje parametra  $\mu_c$  je modifikacija modificirane faktorizacije Choleskog dane u knjizi Gill, Murray i Wright [2]. Najprije se primijeni Gill i Murrayjev algoritam modificirane faktorizacije Choleskog na  $\nabla^2 f(x_c)$ , što rezultira s  $\nabla^2 f(x_c) + D = LL^T$ , pri čemu je  $D$  dijagonalna matrica s nenegativnim elementima na dijagonali, koji su jednaki 0 ako je  $\nabla^2 f(x_c)$  sigurno pozitivno definitna. Ako je  $D = 0$  i  $\mu_c = 0$ , gotovi smo. Inače, ako je  $D \neq 0$ , računamo gornju granicu  $b_1$  na  $\mu_c$  koristeći teorem Gerschgorinovih krugova<sup>4</sup>. Budući da je

$$b_2 = \max_{1 \leq i \leq n} \{d_{ii}\}$$

isto gornja granica na  $\mu_c$ , stavljamo  $\mu_c = \min\{b_1, b_2\}$  i završavamo algoritam računajući faktorizaciju Choleskog  $H_c = \nabla^2 f(x_c) + \mu_c I$ .

Rezultat gornjeg razmatranja je modificirana verzija Newtonove metode:

### Algoritam 1.3.3. Modifikacija Newtonove metode za problem bezuvjetne minimizacije

Za dva puta neprekidnu diferencijabilnu funkciju  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  i  $x_0 \in \mathbb{R}^n$ : kod svake iteracije  $k$ :

Nađi  $\mu \geq 0$  takav da je  $H_k = \nabla^2 f(x_k) + \mu_k I$  pozitivno definitna te nađi dekompoziciju Choleskog od  $H_k$ , gdje je

- $\mu_k = 0$ , ako je  $\nabla^2 f(x_k)$  sigurno pozitivno definitna
- $\mu_k > 0$  je dovoljno velik tako da je  $H_k$  sigurno pozitivno definitna, inače

Riješi:

$$\begin{aligned} H_k s_k^N &= -\nabla f(x_k), \\ x_{k+1} &= x_k + s_k^N. \end{aligned}$$

---

<sup>4</sup>Neka je  $A \in \mathbb{R}^{n \times n}$  simetrična matrica sa svojstvenim vrijednostima  $\lambda_1, \dots, \lambda_n$ . Tada vrijedi  $\max_{1 \leq k \leq n} \lambda_k \leq \max_{1 \leq k \leq n} \{a_{kk} + \sum_{j=1, j \neq k}^n |a_{kj}|\}$

Kada je  $x_0$  dovoljno blizu minimumu  $x_*$  te  $\nabla^2 f(x_*)$  pozitivno definitna i dobro uvjetovana, tada je Algoritam 1.3.3 ekvivalentan Algoritmu 1.3.1, pa prema tome i  $q$ -kvadratno konvergentan. Prednost Algoritma 1.3.3 je u tome što je modificirani Newtonov smjer  $-H_c^{-1}\nabla f(x_c)$ , smjer u kojem se svaka dva puta diferencijabilna funkcija  $f(x)$  inicijalno smanjuje iz bilo koje točke  $x_c$ .

Dvije su važne interpretacije Algoritma 1.3.3 u terminima modela kvadratnog polinoma od  $f(x_c)$ . Algoritam postavlja  $x_+$  jednak minimumu  $x_c + p$  aproksimacije

$$m_c(x_c + p) = f(x_c) + \nabla f(x_c)^T p + \frac{1}{2} p^T H_c p.$$

U Poglavlju 4 će se pokazati da je  $x_+$  i minimum nemodificiranog modela

$$m_c(x_c + p) = f(x_c) + \nabla f(x_c)^T p + \frac{1}{2} p^T \nabla^2 f(x_c) p,$$

uz uvjet

$$\|p\|_2 \leq \|x_+ - x_c\|_2.$$

Obje interpretacije se primjenjuju u objašnjenju glavnog problema modificirane Newtonove metode, a to je da u nekim situacijama može rezultirati prevelikim koracima  $s_c^N$ . Tim problemom ćemo se baviti u narednim poglavljima.

## Poglavlje 2

# Uvod u globalne metode za nelinearne jednadžbe i bezuvjetnu minimizaciju

### 2.1 Kvazi-Newtonov okvir

Osnovna ideja uspješnog nelinearnog algoritma je kombiniranje globalno konvergentne metode sa brzom lokalnom kako bi se iskoristila prednost obje. Budući da je Newtonova metoda, kada su zadovoljeni određeni uvjeti, brza lokalna strategija, kod svake iteracije algoritma poželjno je primijeniti Newtonovu metodu ili njenu modifikaciju. Ukoliko se time postiže zadovoljavajuće rješenje, npr. ako  $f$  opada prilikom minimizacije, koristimo tu metodu. U slučaju da rješenje nije zadovoljavajuće, vraćamo se iteraciji primijenjući globalnu strategiju. Opisan algoritam će uvijek završiti primijenom Newtonove metode blizu rješenja i time zadržati brzu lokalnu konvergenciju. Ukoliko je pritom globalna metoda dobro inkorporirana, algoritam će biti i globalno konvergentan. Takav algoritam naziva se kvazi-Newtonov algoritam.

**Algoritam 2.1.1. Kvazi-Newtonov algoritam za nelinearne jednadžbe ili bezuvjetne minimizacije** (za optimizaciju, zamijenimo  $F(x_k)$  sa  $\nabla f(x_k)$  i  $J_k$  sa  $H_k$ )

Neka je funkcija  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  neprekidno diferencijabilna i  $x_0 \in \mathbb{R}^n$ .

U svakoj iteraciji  $k$ :

1. Ako već nisi, izračunaj  $F(x_k)$ , i donesi odluku - STOP ili nastavi.
2. Izračunaj  $J_k$  da bude  $J(x_k)$  ili njena aproksimacija.
3. Faktoriziraj  $J_k$  i procijeni njezin broj uvjetovanosti. Ako je  $J_k$  loše uvjetovana, ispravno je perturbiraj.
4. Riješi  $J_k s_k^N = -F(x_k)$ .

5. Donesi odluku - primijeni Newtonov korak  $x_{k+1} = x_k + s_k^N$  ili odaberi  $x_{k+1}$  globalnom metodom. Vрати se na korak 1.

Koraci 2., 3. i 4. su već poznati, stoga se u nastavku razmatra 5. korak gornjeg algoritma, odnosno uvjeti odlučivanja primijene lokalne (Newtonove) ili globalne metode te kriterij zaustavljanja (korak 1.), kako bi se na kraju konstruirao kvazi-Newtonov algoritam.

## 2.2 Smjer silaska

Diskusija o globalnim metodama započinje promatrajući bezuvjetan problem minimizacije

$$\min_{x \in \mathbb{R}^n} f : \mathbb{R}^n \longrightarrow \mathbb{R} \quad (2.1)$$

zbog postojanja prirodne globalne metode koja osigurava smanjenje vrijednosti funkcije  $f$  u svakom koraku. Metode za problem minimizacije korespondiraju metodama sustava nelinearnih jednadžbi, osiguravajući da svaki korak smanjuje vrijednost neke norme  $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ . Primjenom norme, problem rješavanja sustava jednadžbi  $F(x) = 0$  transformira se u problem minimizacije funkcije

$$f := \frac{1}{2} \|F\|_2^2 : \mathbb{R}^n \longrightarrow \mathbb{R}, \quad (2.2)$$

pa se najprije razmatraju globalne metode bezuvjetne minimizacije, a kasnije se transformacijom (2.2) primijenjuju na sustav nelinearnih jednadžbi.

Osnovna ideja globalne metode bezuvjetne minimizacije je sljedeća: uzmi korak silaska funkcije  $f$ , odnosno odaberi smjer  $p$  od trenutne točke  $x_c$  u kojoj se  $f$  inicijalno smanjuje i novu točku  $x_+$  u tom smjeru počevši od  $x_c$  tako da je  $f(x_+) < f(x_c)$ . Budući da je gradijent funkcije smjer maksimalnog rasta funkcije  $f$ , smjer  $-\nabla f(x)$  je tada smjer maksimalnog silaska. Prema tome,  $p$  je smjer silaska iz  $x_c$ , ako je derivacija duž vektora  $p$  od  $f$  u točki  $x_c$  negativna, tj. ako vrijedi

$$\nabla f(x_c)^T p < 0 \quad (2.3)$$

Vrijedi li (2.3), osigurano je da za dovoljno malen  $\delta > 0$  vrijedi  $f(x_c + \delta p) < f(x_c)$ .

Promotri li se Newtonova metoda, za minimizaciju se uzima Newtonov smjer  $s^N = -H_c^{-1} \nabla f(x_c)$ , gdje je matrica  $H_c$  jednaka  $\nabla^2 f(x_c)$  ili njenoj aproksimaciji. Postavlja se pitanje: je li Newtonov smjer smjer silaska? Prema (2.3) jest ako i samo ako je

$$\nabla f(x_c)^T s^N = -\nabla f(x_c)^T H_c^{-1} \nabla f(x_c) < 0,$$

što vrijedi ako je  $H_c^{-1}$ , ili ekvivalentno  $H_c$ , pozitivno definitna.

Pozitivna definitnost osigurava postojanje jedinstvenog minimuma kvadratanog modela i da rezultirajući Newtonov smjer bude smjer silaska za konkretan problem, što znači da će vrijednost funkcije opadati za dovoljno malen korak u Newtonovom smjeru.

Iduće pitanje koje se postavlja je: sve dok uzimamo korak u smjeru silaska, koji je smjer  $p$  u kojem  $f$  najbrže opada? Kako je derivacija duž vektora  $p$  od  $f$  proporcionalna duljini vektora  $p$ , definiramo smjer vektora  $p$  kao  $\frac{p}{\|p\|}$  te u nastavku za smjer  $p$  uvijek pretpostavljamo da je provedena ova normalizacija. Sada najbrži smjer silaska za danu normu  $l_2$

$$\min_{p \in \mathbb{R}^n} \nabla f(x)^T p \quad \text{uz uvjet } \|p\| = 1,$$

ima rješenje  $p = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2}$ , pa je najbrži smjer silaska upravo negativan smjer gradijenta funkcije  $f$ .

### Algoritam 2.2.1. Metoda najbržeg silaska

Neka je funkcija  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  neprekidno diferencijabilna i  $x_0 \in \mathbb{R}^n$ . Za svaku iteraciju  $k$ : nađi najmanju točku funkcije  $f$  u smjeru  $-\nabla f(x_k)$  iz točke  $x_k$ , tj. nađi  $\lambda_k$  koji rješava:

$$\min_{\lambda_k > 0} f(x_k - \lambda_k \nabla f(x_k))$$

$$x_{k+1} := x_k - \lambda_k \nabla f(x_k)$$

Klasičan algoritam minimizacije baziran je na najbržem smjeru opadanja, ali kao računska metoda nije dovoljno efikasan. Pokazuje se da pod blagim uvjetima konvergira lokalnom minimumu ili sedlastoj točki (Goldstein [3]), ali konvergencija je samo linearna i ponekad vrlo spora.

**Primjer 2.2.2.** Neka je  $f(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$ , pozitivno definitan kvadratni polinom s minimumom u  $x_* = (0, 0)^T$  i

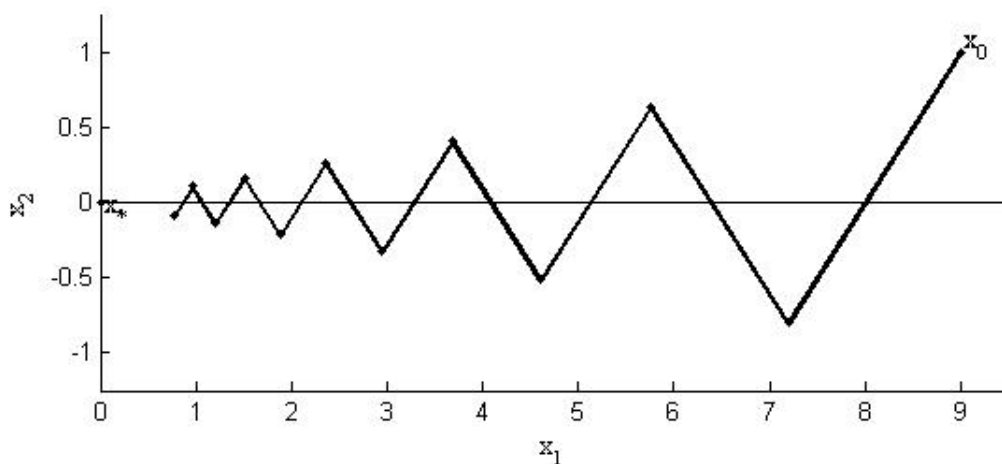
$$\nabla^2 f(x) = \begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix}, \quad \forall x.$$

Korak iteracije metode najbržeg silaska dan je sa

$$x_{k+1} = x_k - \frac{g^T g}{g^T \nabla^2 f(x_k) g} g, \quad (2.4)$$

gdje je  $g = \nabla f(x_k)$ . Primjenom (2.4) niz točaka generiran Algoritmom 2.2.1 za danu  $f(x)$  i  $x_0 = (9, 1)^T$  jednak je

$$x_k = \begin{pmatrix} 9 \\ (-1)^k \end{pmatrix} 0.8^k \quad k = 1, 2, \dots \quad (2.5)$$



Slika 2.1: Metoda najbržeg silaska na loše skaliranom kvadratnom polinomu

Niz  $x_k$  dan u (2.5) prikazan je na Slici 2.1. Vidljivo je da metodi najbržeg silaska treba vremena da iskonvergira. Zna se da je rješenje  $x_* = (0, 0)$ , ali čak i nakon 55 iteracija dobivena preciznost je relativno malena. Vidljivo je da vrijednost funkcije  $f$  ide prema  $x_* = (0, 0)$ , odnosno da konvergira, ali konvergencija je vrlo spora. S druge strane, vidjeli smo u Poglavlju 1 da Newtonova metoda već u 6 koraka daje jako dobru aproksimaciju.

Dakle, metoda najbržeg silaska se u praksi ne bi smjela često koristiti, budući da će kvazi-Newtonov algoritam biti efikasniji. Međutim, Newtonova metoda ponekad nije pouzdana, npr. kada je poželjno napraviti puno manji korak od Newtonovog. U tim slučajevima se razmatra primjena metode najbržeg silaska.

Nadalje, metoda najbržeg silaska je osjetljiva na promjene u skali od  $x$ , stoga se u algoritmu Poglavlja 4, odgovarajuća implementacija množi s dijagonalnom matricom skaliranja, razmatranom u Poglavlju 6.1.

## Poglavlje 3

# Line search - pretraživanje po pravcu

Metoda koja se nadovezuje na razmatranje metode smjera silaska je metoda line search (metoda pretraživanja po pravcu).

### Ideja line search algoritma:

Za dani smjer silaska  $p$ , uzmi korak u tom smjeru koji rezultira “prihvatljivim”  $x_{k+1}$ , tj. kod iteracije  $k$ :

izračunaj smjer silaska  $p_k$ ,

definiraj  $x_{k+1} := x_k + \lambda_k p_k$  za neki  $\lambda_k$  za koji je  $x_{k+1}$  prihvatljiva sljedeća iteracija. (3.1)

Izraz “line search” odnosi se na postupak biranja  $\lambda_k$  u (3.1). Ideja je uzeti bilo koji prihvatljiv korak i zadržati ga. Ukoliko korak nije prihvatljiv, pokušati ponovo, bez puno izračuna. Pri odabiru prihvaćanja koraka, budući da se traži minimum funkcije, zahtijeva se da vrijednost funkcije opada, tj. da bude zadovoljeno

$$f(x_{k+1}) < f(x_k) \quad (3.2)$$

Dakle, za danu funkciju  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , trenutnu iteraciju  $x_k \in \mathbb{R}^n$  i smjer silaska  $p_k \in \mathbb{R}^n$ ,  $\nabla f(x_k)^T p_k < 0$ , nađi

$$x_{k+1} = x_k + \lambda_k p_k$$

gdje je  $\lambda_k$  takav da vrijedi (3.2).

Iako se uvjet (3.2) čini jednostavnim, lako se pokaže da ne garantira konvergenciju niza  $x_k$  u minimum funkcije  $f$ . Naime, niz ponavljanja može zadovoljavati uvjet (3.2), ali podbaciti u konvergenciji u minimum, što će biti prikazano u sljedeća dva primjera.

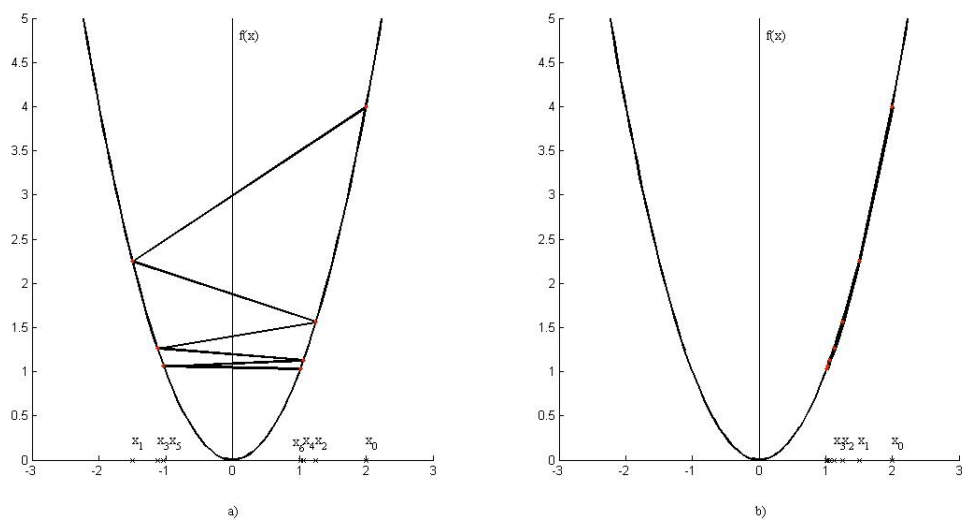


**Primjer 3.0.1.** Neka je  $f(x) = x^2$ ,  $x_0 = 2$ . Ako se izabere  $p_k = -\text{sgn}(x_k)$ , gdje je

$$\text{sgn}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0, \end{cases}$$

tj.  $p_k = (-1)^{k+1}$  i korak  $\lambda_k = 2 + 3(2^{-(k+1)})$ , tada je  $x_k = \{2, -\frac{3}{2}, \frac{5}{4}, -\frac{9}{8}, \dots\} = \{(-1)^k(1 + 2^{-k})\}$ ,  $p_k$  je smjer silaska i  $f$  monotono opada,  $f(x_{k+1}) < f(x_k)$ . [Slika 3.1 a)] Međutim,  $\lim_{k \rightarrow \infty} x_k$  ne postoji, štoviše  $\{x_k\}$  ima dva gomilišta  $\pm 1$ , pa ne konvergira.

**Primjer 3.0.2.** Neka je funkcija ista kao u primjeru 3.0.1, samo neka je sada  $p_k = -1$  i  $\lambda_k = 2^{-k+1}$ . Tada je  $\{x_k\} = \{2, \frac{3}{2}, \frac{5}{4}, \frac{9}{8}, \dots\} = \{1 + 2^{-k}\}$ ,  $p_k$  je smjer silaska, vrijedi  $f(x_{k+1}) < f(x_k)$  i  $\lim_{k \rightarrow \infty} x_k = 1$ . Dakle, u ovom slučaju  $\{x_k\}$  konvergira, ali ne u minimum funkcije  $f$ . [Slika 3.1 b)]



Slika 3.1: Monotono opadajući nizovi iteracija koji ne konvergiraju prema minimumu

Inutivno, metoda u Primjeru 3.0.1 ne konvergira jer je smjer silaska zasnovan na Taylorovom teoremu aproksimacije funkcije linearnim modelom koja vrijedi samo lokalno, tj. pretpostavlja samo malene korake. Ukoliko je smanjenje vrijednosti funkcije vrlo malo u odnosu na duljinu koraka, konvergencija neće biti osigurana.

U Primjeru 3.0.2, napretkom iteracija, koraci su sve manji, tj. premali su u odnosu na početnu stopu smanjenja funkcije  $f$ , što dovodi do toga da algoritam ne napreduje.

Dakle, odabir bilo kojeg koraka, koji rezultira smanjenjem vrijednosti funkcije, nije uvijek efikasan. Ako je korak predug, iako rezultira smanjenjem vrijednosti funkcije, nije konzistentan s Taylorovim teoremom i nema garancije konvergencije. S druge strane, ako je korak prekratak, algoritam će vrlo vjerojatno doći do točke nakon koje više ne napreduje prema rješenju.

Iduće potpoglavlje daje nužne uvjete za postizanje konvergencije.

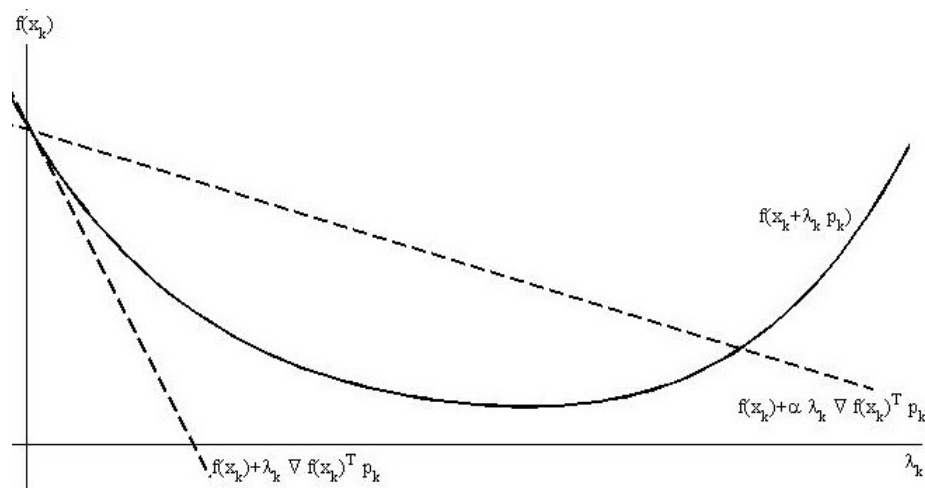
### 3.1 Wolfe-ovi uvjeti i konvergentni rezultati

#### 3.1.1 Wolfe-ovi uvjeti

U problemu bezuvjetne minimizacije, Wolfe-ovi uvjeti (Wolfe [4] i [5]), zasnovani na radu Armijo [6] i Goldstein [3], su skupovi nejednakosti za izvođenje neegzaktne line search algoritma, posebno u kvazi-Newtonovim metodama.

Prvi Wolfe-ov uvjet, poznat kao Armijo pravilo, zahtijeva da prosječna stopa smanjenja  $f(x_k)$  do  $f(x_{k+1})$  bude barem neki propisani dio početne stope smanjenja u tom smjeru, tj. potrebno je odabrati  $\alpha \in (0, 1)$  i  $\lambda_k > 0$  koji zadovoljava

$$f(x_k + \lambda_k p_k) \leq f(x_k) + \alpha \lambda_k \nabla f(x_k)^T p_k \quad (3.3)$$



Slika 3.2: Dopustive vrijednosti  $\lambda$  za uvjet (3.3)

Zahtijeva se dovoljan pad funkcije i da stopa smanjenja bude proporcionalna nagibu funkcije  $\nabla f(x_k)^T p_k$ , proporcionalna koraku  $\lambda_k$  te parametru  $\alpha$  koji je striktno između 0 i 1

čime se izolira uzimanje predugih koraka. Lako se pokaže da, kada je  $\alpha$  bliže 0, koraci su dulji, a kada je bliže 1 koraci su kraći.

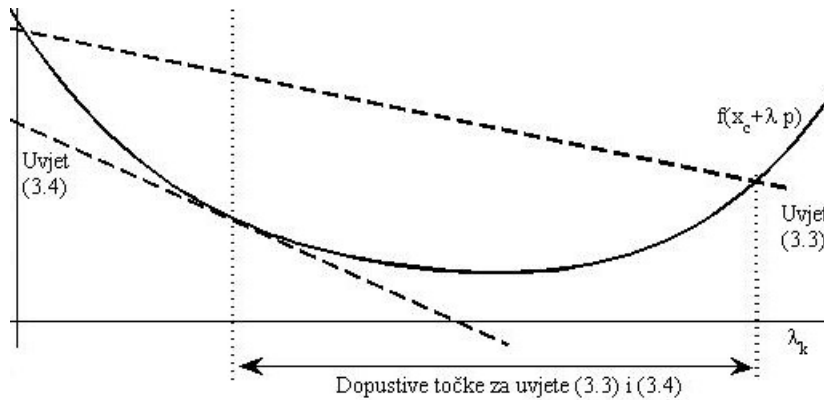
Drugi Wolfe-ov uvjet, poznat kao uvjet krivulje, zahtijeva da stopa smanjenja funkcije  $f$  u smjeru silaska  $p$  u točki  $x_{k+1}$  bude veća od nekog propisanog razlomka stope smanjenja u smjeru  $p$  u točki  $x_k$ , tj.

$$\nabla f(x_{k+1})^T p_k := \nabla f(x_k + \lambda_k p_k)^T p_k \geq \beta \nabla f(x_k)^T p_k \quad (3.4)$$

za konstantu  $\beta \in (0, 1)$  [Slika 3.3].

Nametnut je dovoljan napredak iteracije, odnosno relativni rast derivacije duž vektora za najmanje  $\beta$ ,  $0 < \beta < 1$ , čime se izolira uzimanje prekratkih koraka. Nadalje, lako se pokaže da, ako je  $\beta$  bliže 0, nova vrijednost derivacije duž vektora će biti blizu 0, tj. korak će biti dulji. Ako je  $\beta$  bliže 1, koraci će biti manji i nova vrijednost će biti blizu prethodne, što daje više fleksibilnosti algoritmu.

Uvjet  $\beta > \alpha$ , tj. uvjet  $0 < \alpha < \beta < 1$ , omogućava istovremenu zadovoljenost oba Wolfe-ova uvjeta (3.3) i (3.4). U praksi, uvjet (3.4) nije potreban, jer primjena *backtracking* algoritma izbjegeva pretjerano male korake.



Slika 3.3: Oba Wolfe-ova uvjeta

**Primjer 3.1.1.** Neka je  $f(x_1, x_2) = x_1^4 + x_1^4 + x_2^4$ ,  $x_k = (1, 1)^T$ ,  $p_k = (-3, -1)^T$  i neka je  $\alpha = 0.1$  u (3.3) i  $\beta = 0.5$  u (3.4). Kako je

$$\nabla f(x_k)^T p_k = (6, 2)(-3, -1)^T = -20 < 0,$$

$p_k$  je smjer silaska. Neka je sada  $x(\lambda_k) = x_k + \lambda_k p_k$ . Ako je  $\lambda_k = 1$ , tada je  $x_{k+1} = x(1) = (-2, 0)^T$  i

$$\nabla f(x_{k+1})^T p_k = (-36, 0)(-3, -1)^T = 108 > -10 = \beta \nabla f(x_k)^T p_k.$$

Prema tome,  $x_{k+1}$  zadovoljava (3.4), ali ne zadovoljava (3.3)

$$f(x_{k+1}) = 20 > f(x_k) + \alpha \nabla f(x_k)^T p_k.$$

Slično se provjeri da  $x_{k+1} = x(0.1) = (0.7, 0.9)^T$  zadovoljava (3.3), ali ne i (3.4) i da  $x_{k+1} = x(0.5) = (0.5, 0.5)^T$  zadovoljava i (3.3) i (3.4). Točke  $x(1)$ ,  $x(0.1)$  i  $x(0.5)$  korespondiraju točkama desno, lijevo i u “dopuštenom području” Slike 3.3 respektivno.

### 3.1.2 Konvergencija

Koristeći Wolfe-ove uvjete, mogu se dokazati jaki rezultati koji osiguravaju kreiranje globalno konvergentnih algoritama, kao i, za većinu funkcija, brzo lokalno konvergentnih algoritama.

Prvi takav rezultat prikazuje da za bilo koji dani smjer  $p_k$ , postoji točka  $x_k + \lambda_k p_k$  koja zadovoljava uvjete (3.3) i (3.4).

**Teorem 3.1.2.** *Neka je  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  neprekidno diferencijabilna na  $\mathbb{R}^n$ . Neka  $x_k \in \mathbb{R}^n$  i  $p_k \in \mathbb{R}^n$  zadovoljavaju  $\nabla f(x_k)^T p_k < 0$  te pretpostavimo da je skup  $\{f(x_k + \lambda p_k) \mid \lambda > 0\}$  omeđen odozdo. Tada ako je  $0 < \alpha < \beta < 1$ , postoje  $\lambda_u > \lambda_l > 0$  takvi da  $x_{k+1} = x_k + \lambda_k p_k$  zadovoljava (3.3) i (3.4), ako je  $\lambda_k \in (\lambda_l, \lambda_u)$ .*

*Dokaz.* Kako je  $\alpha < 1$ ,  $f(x_k + \lambda p_k) < f(x_k) + \alpha \lambda \nabla f(x_k)^T p_k$  za sve  $\lambda > 0$  dovoljno mali i kako je  $f(x_k + \lambda p_k)$  također ograničena odozdo, postoji neki mali pozitivni  $\hat{\lambda}$  takav da

$$f(\hat{x}) = f(x_k + \hat{\lambda} p_k) = f(x_k) + \alpha \hat{\lambda} \nabla f(x_k)^T p_k. \quad (3.5)$$

Tako svaki  $x \in (x_k, \hat{x})$  zadovoljava (3.3). Po teoremu srednje vrijednosti, postoji  $\bar{\lambda} \in (0, \hat{\lambda})$  takav da

$$\begin{aligned} f(\hat{x}) - f(x_k) &= \nabla f(x_k + \bar{\lambda} p_k)^T (\hat{x} - x_k) \\ &= \nabla f(x_k + \bar{\lambda} p_k)^T \hat{\lambda} p_k, \end{aligned} \quad (3.6)$$

i tako od (3.5) i (3.6) dobijemo

$$\nabla f(x_k + \bar{\lambda} p_k)^T p_k = \alpha \nabla f(x_k)^T p_k > \beta \nabla f(x_k)^T p_k \quad (3.7)$$

budući da je  $\alpha < \beta$  i  $\nabla f(x_k)^T p_k < 0$ . Po neprekidnosti od  $\nabla f$ , (3.7) i dalje vrijedi za  $\lambda$  u nekim intervalima  $(\lambda_l, \lambda_u)$  oko  $\bar{\lambda}$ . Dakle, ograničimo li  $(\lambda_l, \lambda_u)$  da bude unutar  $(0, \hat{\lambda})$ ,  $x_{k+1} = x_k + \lambda_k p_k$  zadovoljava (3.3) i (3.4) za svaki  $\lambda_k \in (\lambda_l, \lambda_u)$ .  $\square$

Pokazuje se da, ako bilo koji niz  $x_k$ , generiran da zadovoljava  $\nabla f(x_k)^T (x_{k+1} - x_k) < 0$ , (3.3) i (3.4), osim ako kut između  $\nabla f(x_k)$  i  $(x_{k+1} - x_k)$  ne konvergira prema kutu od  $90^\circ$  kada  $k \rightarrow \infty$ , tada ili gradijent konvergira u 0 ili je  $f$  neograničena odozdo ili oboje. Idući teorem pokazuje, kada se kut između  $\nabla f(x_k)$  i  $(x_{k+1} - x_k)$  približava kutu od  $90^\circ$ , da se to algoritmom može spriječiti.

**Teorem 3.1.3.** *Neka je  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  neprekidno diferencijabilna na  $\mathbb{R}^n$  i pretpostavimo da postoji  $\gamma \geq 0$  takav da vrijedi*

$$\|\nabla f(z) - \nabla f(x)\|_2 \leq \gamma \|z - x\|_2 \quad (3.8)$$

za svaki  $x, z \in \mathbb{R}^n$ . Tada za bilo koji  $x_0 \in \mathbb{R}^n$ , ili je  $f$  neograničena odozdo ili postoji niz  $\{x_k\}$ ,  $k = 0, 1, \dots$  koji zadovoljava (3.3) i (3.4) te

$$\nabla f(x_k)^T s_k < 0 \quad (3.9)$$

ili

$$\nabla f(x_k) = 0 \text{ i } s_k = 0,$$

za svaki  $k \geq 0$ , gdje je

$$s_k := x_{k+1} - x_k.$$

Nadalje, za bilo koji takav niz, ili je

(a)  $\nabla f(x_k) = 0$  za neki  $k \geq 0$ , ili je

(b)  $\lim_{k \rightarrow \infty} f(x_k) = -\infty$  ili je

(c)  $\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T s_k}{\|s_k\|_2} = 0$ .

*Dokaz.* Za svaki  $k$ , ako je  $\nabla f(x_k) = 0$ , tada vrijedi (a) i niz je naknadno konstantan. Ako je  $\nabla f(x_k) \neq 0$ , tada postoji  $p_k$  — npr.  $p_k = -\nabla f(x_k)$  — takav da je  $f(x_k)^T p_k < 0$ . Prema Teoremu 3.1.2, ili je  $f$  neograničena odozdo, ili postoji  $\lambda_k > 0$  takav da  $x_{k+1} = x_k + \lambda_k p_k$  zadovoljava (3.3) i (3.4). Kako bismo pojednostavili notaciju, pretpostavit ćemo da je  $\|p_k\|_2 = 1$  tako da je  $\lambda_k = \|s_k\|_2$ , što ne predstavlja gubitak općenitosti (generalizacije).

Dosad vidimo da je  $f$  ili neograničeno odozdo ili postoji  $\{x_k\}$  i  $\{s_k\}$  ili zadovoljava (a) ili je  $s_k \neq 0$  za svaki  $k$ . Sada moramo pokazati da ako nijedan član iz  $\{s_k\}$  nije nula, tada mora vrijediti (b) ili (c). Bit će korisno imati

$$\sigma_k := \frac{\nabla f(x_k)^T s_k}{\|s_k\|_2}.$$

Iz (3.3) i  $\lambda_k \sigma_k < 0$  za svaki  $k$ , za neki  $j > 0$  vrijedi

$$\begin{aligned}
f(x_j) - f(x_0) &= \sum_{k=0}^{j-1} f(x_{k+1}) - f(x_k) \\
&\leq \sum_{k=0}^{j-1} \alpha \nabla f(x_k)^T s_k \\
&= \alpha \sum_{k=0}^{j-1} \lambda_k \sigma_k < 0
\end{aligned}$$

Stoga je ili

$$\lim_{j \rightarrow \infty} f(x_j) = -\infty \quad \text{ili je} \quad - \sum_{k=0}^{\infty} \lambda_k \sigma_k < \infty$$

konvergentno.

U prvom slučaju je (b) istinito i tada smo gotovi, pa promotrimo drugi slučaj. Ustvari, zaključujemo da vrijedi  $\lim_{k \rightarrow \infty} \lambda_k \sigma_k = 0$ . Sada želimo zaključiti da je  $\lim_{k \rightarrow \infty} \sigma_k = 0$ , pri čemu trebamo koristiti uvjet (3.4), obzirom da je napomenuto da se osigura da koraci ne postanu premali.

Za svaki  $k$  vrijedi

$$\nabla f(x_{k+1})^T s_k \geq \beta \nabla f(x_k)^T s_k$$

kao i

$$[\nabla f(x_{k+1}) - \nabla f(x_k)]^T s_k \geq (\beta - 1) \nabla f(x_k)^T s_k,$$

prema (3.9) i  $\beta < 1$ . Primijenjujući Cauchy-Schwarzovu nejednakost i (3.8) na lijevu stranu posljednje jednadžbe i koristeći definiciju od  $\sigma_k$ , dobivamo

$$0 < (\beta - 1) \lambda_k \sigma_k \leq \gamma \|s_k\|^2 = \gamma \lambda_k^2,$$

pa vrijedi

$$\lambda_k \geq \frac{\beta - 1}{\gamma} \sigma_k > 0$$

i

$$\lambda_k \sigma_k \leq \frac{\beta - 1}{\gamma} \sigma_k^2 < 0.$$

Prema tome vrijedi

$$0 = \lim_{k \rightarrow \infty} \lambda_k \sigma_k \leq \frac{\beta - 1}{\gamma} \lim_{k \rightarrow \infty} \sigma_k^2 \leq 0$$

što pokazuje da vrijedi

$$\lim_{k \rightarrow \infty} \sigma_k = 0$$

[tj., (c) je istinito] i time je dokaz potpun.  $\square$

Teorem 3.1.3 je potpuno neovisan od metode odabira smjera silaska ili duljine koraka i lako se primijenjuje na bilo koji line search algoritam te daje dovoljne uvjete globalne konvergencije za svaki optimizacijski algoritam.

Sljedeći teorem, Dennis i Móre [7], pokazuje da će globalna metoda dozvoljavati puni kvazi-Newtonov korak  $x_{k+1} := x_k - H_k^{-1} \nabla f(x_k)$  u blizini minimuma funkcije  $f$  sve dok je  $-H_k^{-1} \nabla f(x_k)$  dovoljno blizu Newtonovom koraku.

**Teorem 3.1.4.** *Neka je  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  dva puta neprekidno diferencijabilna na otvorenom konveksnom skupu  $D$  i pretpostavimo da je  $\nabla^2 f \in Lip_\gamma(D)$ . Neka je niz  $\{x_k\}$  generiran sa  $x_{k+1} = x_k + \lambda_k p_k$ , gdje je  $\nabla f(x_k)^T p_k < 0$  za svaki  $k$  i  $\lambda_k$  izabran da zadovoljava (3.3) i (3.4) sa  $\alpha < \frac{1}{2}$ . Ako  $\{x_k\}$  konvergira prema točki  $x_* \in D$  u kojoj je  $\nabla^2 f(x_*)$  pozitivno definitna i ako vrijedi*

$$\lim_{k \rightarrow \infty} \frac{\|\nabla f(x_k) + \nabla^2 f(x_k) p_k\|_2}{\|p_k\|_2} = 0, \quad (3.10)$$

tada postoji indeks  $k_0 \geq 0$  takav da je, za sve  $k \geq k_0$ ,  $\lambda_k = 1$  dopustivo. Nadalje,  $\nabla f(x_*) = 0$  i ako je  $\lambda_k = 1$  za sve  $k \geq k_0$ , tada niz  $\{x_k\}$  konvergira  $q$ -superlinearno u  $x_*$ .

*Dokaz.* Dokaz je zapravo samo generalizacija jednostavne tvrdnje da ako je  $f(x)$  kvadratno pozitivno definitna i  $p_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$ , tada je  $\nabla f(x_k)^T p_k < 0$ ,  $x_{k+1} = x_k + p_k$  zadovoljava (3.3) za bilo koji  $\alpha \leq \frac{1}{2}$  i (3.4) za bilo koji  $\beta \geq 0$ . Stavimo

$$\rho_k = \frac{\|\nabla f(x_k) + \nabla^2 f(x_k) p_k\|}{\|p_k\|}, \quad (3.11)$$

pri čemu  $\|\cdot\|$  označava  $\|\cdot\|_2$  tokom cijelog dokaza. Prvo dokazujemo da vrijedi

$$\lim_{k \rightarrow \infty} \|p_k\| = 0.$$

Prema poznatom argumentu, ako je  $\bar{x}$  dovoljno blizu  $x_*$ , tada postoji pozitivno definitna matrica  $\nabla^2 f(\bar{x})^{-1}$  i ako je  $\mu^{-1} = \|\nabla^2 f(x_*)^{-1}\|$ , tada vrijedi  $\frac{1}{2}\mu^{-1} \leq \|\nabla^2 f(\bar{x})^{-1}\| \leq 2\mu^{-1}$ . Prema tome, kako je

$$-\frac{\nabla f(x_k)^T p_k}{\|p_k\|} = \frac{p_k^T \nabla^2 f(x_k) p_k}{\|p_k\|} - \frac{(\nabla f(x_k) + \nabla^2 f(x_k) p_k)^T p_k}{\|p_k\|}$$

i  $p_k^T \nabla^2 f(x_k) p_k > \frac{1}{2}\mu \|p_k\|^2$ , (3.11) i (3.10) pokazuju da za dovoljno velik  $k$ , vrijedi

$$-\frac{\nabla f(x_k)^T p_k}{\|p_k\|} \geq \left(\frac{1}{2}\mu - \rho_k\right) \|p_k\| \geq \frac{1}{4}\mu \|p_k\|. \quad (3.12)$$

Iz Teorema 3.1.3 znamo da vrijedi

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T p_k}{\|p_k\|} = 0,$$

a (3.12) implicira da vrijedi

$$\lim_{k \rightarrow \infty} \|p_k\| = 0.$$

Sada dokazujemo da  $\lambda_k = 1$  zadovoljava (3.3), za sve  $k$ -ove veće ili jednake nekom  $k_0$ . Iz teorema srednje vrijednosti, za neke  $\bar{x}_k \in [x_k, x_k + p_k]$ ,

$$f(x_k + p_k) - f(x_k) = \nabla f(x_k)^T p_k + \frac{1}{2} p_k^T \nabla^2 f(\bar{x}_k) p_k,$$

ili

$$\begin{aligned} f(x_k + p_k) - f(x_k) - \frac{1}{2} \nabla f(x_k)^T p_k &= \frac{1}{2} (\nabla f(x_k) + \nabla^2 f(\bar{x}_k) p_k)^T p_k \\ &= \frac{1}{2} (\nabla f(x_k) + \nabla^2 f(x_k) p_k)^T p_k + \frac{1}{2} p_k^T [\nabla^2 f(\bar{x}_k) - \nabla^2 f(x_k)] p_k \\ &\leq \frac{1}{2} (\rho_k + \gamma \|p_k\|) \|p_k\|^2 \end{aligned} \tag{3.13}$$

zbog (3.11) i Lipschitz neprekidnosti od  $\nabla^2 f$ . Sada odaberemo  $k_0$  tako da za  $k \geq k_0$  vrijedi (3.12) i

$$\rho_k + \gamma \|p_k\| \leq \frac{1}{4} \mu \cdot \min(\beta, 1 - 2\alpha). \tag{3.14}$$

Ako je  $k \geq k_0$ , tada  $\lambda_k = 1$  zadovoljava (3.3), jer iz (3.13), (3.12) i (3.14) slijedi

$$\begin{aligned} f(x_k + p_k) - f(x_k) &\leq \frac{1}{2} \nabla f(x_k)^T p_k + \frac{1}{8} \mu \cdot (1 - 2\alpha) \|p_k\|^2 \\ &\leq \frac{1}{2} (1 - (1 - 2\alpha)) \nabla f(x_k)^T p_k \\ &= \alpha \nabla f(x_k)^T p_k. \end{aligned}$$

Kako bismo pokazali da je (3.4) zadovoljeno sa  $\lambda_k = 1$  za  $k \geq k_0$ , ponovo koristimo teorem srednje vrijednosti kako bismo za neki  $z_k \in (x_k, x_k + p_k)$  dobili da vrijedi

$$\begin{aligned} \nabla f(x_k + p_k)^T p_k &= (\nabla f(x_k) + \nabla^2 f(z_k) p_k)^T p_k \\ &= (\nabla f(x_k) + \nabla^2 f(x_k) p_k)^T p_k + p_k^T (\nabla^2 f(z_k) - \nabla^2 f(x_k)) p_k \end{aligned}$$



tako da je

$$\begin{aligned} |\nabla f(x_k + p_k)^T p_k| &\leq \rho_k \|p_k\|^2 + \gamma \|p_k\|^3 \\ &\leq \frac{\mu\beta}{4} \|p_k\|^2 \\ &\leq -\beta \nabla f(x_k)^T p_k \end{aligned}$$

pritom koristeći (3.11), Lipschitz neprekidnost od  $\nabla^2 f$ , (3.14) i (3.12). Rezultat toga je  $\nabla f(x_k + p_k)^T p_k \geq \beta \nabla f(x_k)^T p_k$  za  $k \geq k_0$ . Prema tome,  $\lambda_k = 1$  naposljetku zadovoljava (3.3) i (3.4), pa je i dopustiv, što je posljedica od (3.10) i  $\lim_{k \rightarrow \infty} p_k = 0$  tako da  $\nabla f(x_*) = 0$ . Ostaje još samo dokaz  $q$ -superlinearne konvergencije ako je izabran  $\lambda_k = 1$  za sve osim za konačno mnogo uvjeta.  $\square$

Teoremi 3.1.3 i 3.1.4 iskazuju da ako je funkcija  $f$  ograničena odozdo, tada niz  $\{x_k\}$  generiran bilo kojim algoritmom, koji primijenjuje korak silaska, čiji je kut s gradijentom ograničen sa  $90^\circ$ , i zadovoljava uvjete (3.3) i (3.4), zadovoljava

$$\lim_{k \rightarrow \infty} \nabla f(x_k) = 0.$$

Ako takav algoritam prvo uzme Newtonov korak ili kvazi-Newtonov korak kod svake iteracije, tada će  $\{x_k\}$  također konvergirati  $q$ -kvadratno ili  $q$ -superlinearno lokalnom minimumu  $x_*$ , ako je bilo koji  $x_k$  dovoljno blizu  $x_*$  te ako je zadovoljena pretpostavka lokalne konvergencije Newtonove ili kvazi Newtonove metode.

## 3.2 Odabir koraka pomoću backtrackinga

Preostaje odrediti način odabira duljine koraka  $\lambda_k$  u line search algoritmu, pri čemu se koristi metoda *backtrackinga*. U kvazi-Newtonovom algoritmu najprije se ispituje prihvaćanje Newtonove metode, a zatim neke od globalnih metoda. Budući da je Newtonova metoda, kada je  $\nabla^2 f(x_k)^{-1}$  pozitivno definitna matrica, zapravo metoda silaska za  $\lambda_k = 1$ , backtrack line search algoritam također počinje sa  $\lambda_k = 1$  i ako  $x_k + p_k$  nije prihvatljivo, primjenjuje backtrack - smanji  $\lambda_k$ , sve dok ne nađe prihvatljivi  $x_k + \lambda_k p_k$ .

### Algoritam 3.2.1. Backtracking line search

Neka je  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  neprekidno diferencijabilna funkcija,  $x_k \in \mathbb{R}^n$  točka  $k$ -te iteracije i  $\nabla f(x_k) \in \mathbb{R}^n$  pripadni gradijent te  $p_k \in \mathbb{R}^n$  dani smjer silaska. Za dani  $\alpha \in (0, \frac{1}{2})$ , nađi  $x_{k+1} = x_k + \lambda p_k$ ,  $\lambda \in (0, 1)$  takav da vrijedi  $f(x_{k+1}) \leq f(x_k) + \alpha \lambda \nabla f(x_k)^T p_k$ , tj.

Za dani  $\alpha \in (0, \frac{1}{2})$ ,  $0 < l < u < 1$

$\lambda_k = 1$ ;

sve dok je  $f(x_k + \lambda_k p_k) > f(x_k) + \alpha \lambda_k \nabla f(x_k)^T p_k$ , radi sljedeće:

$\lambda_k := \rho \lambda_k$  za neke  $\rho \in [l, u]$ ;

( $\rho$  se nanovo bira svaki put prilikom line search-a)

$x_{k+1} := x_k + \lambda_k p_k$ .

U praksi se većinom uzima vrlo malen  $\alpha$ , ovdje se koristi  $\alpha = 10^{-4}$ , pa preostaje odrediti metodu smanjenja  $\lambda_k$ , odnosno odabir  $\rho$ .

U tu svrhu definira se jednodimenzionalna funkcija  $h$

$$h(\lambda) = f(x_k + \lambda p_k).$$

Za primjenu backtrackinga se koristi posljednja informacija o funkciji  $h$  na temelju koje se modelira kvadratni, odnosno kubični, jednodimenzionalni polinom za traženje minimuma  $\lambda$  što predstavlja sljedeću vrijednost  $\lambda_k$  u Algoritmu 3.2.1.

Poznate su informacije o funkciji  $h$  za  $\lambda = 0$

$$h(0) = f(x_k) \quad \text{i} \quad h'(0) = \nabla f(x_k)^T p_k \quad (3.15)$$

i za početni  $\lambda_k = 1$ , nakon računanja  $f(x_k + p_k)$ ,

$$h(1) = f(x_k + p_k). \quad (3.16)$$

Kada  $f(x_k + p_k)$  ne zadovoljava (3.3), funkcija  $h$  se modelira jednodimenzionalnim kvadratnim polinomom koji zadovoljava (3.15) i (3.16),

$$m_q(\lambda) = [h(1) - h(0) - h'(0)]\lambda^2 + h'(0)\lambda + h(0),$$

pri čemu se računa točka  $\hat{\lambda}$  za koju je  $m'_q(\hat{\lambda}) = 0$ , tj.

$$\hat{\lambda} = \frac{-h'(0)}{2[h(1) - h(0) - h'(0)]} \quad (3.17)$$

Obzirom da nije zadovoljeno Armijo pravilo, vrijedi  $h(1) > h(0) + h'(0)$ , i kako je  $h'(0) < 0$ , kao derivacija duž vektora  $p_k$ , pri čemu je  $p_k$  smjer silaska, vrijedi sljedeće

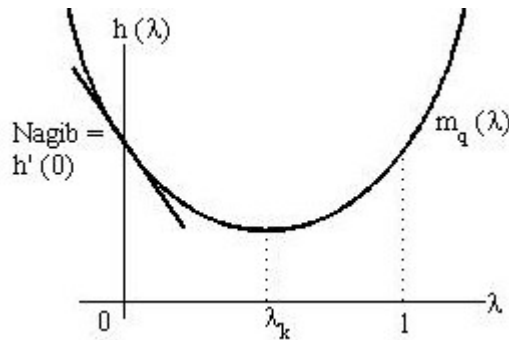
$$m''_q(\hat{\lambda}) = 2[h(1) - h(0) - h'(0)] > 0,$$

tj.  $\hat{\lambda}$  minimizira funkciju  $m_q(\lambda)$ . Također, kako je  $h'(0) < 0$ , slijedi da je  $\hat{\lambda} > 0$ , pa se  $\hat{\lambda}$  uzima kao nova vrijednost  $\lambda_k$ .

Nadalje, može se primijetiti da obzirom na  $h(1) > h(0) + \alpha h'(0)$ , vrijedi

$$\hat{\lambda} < \frac{1}{2(1 - \alpha)}.$$

Kako je  $\alpha \in (0, \frac{1}{2})$ , ako je  $h(1) \geq h(0)$ , tada je  $\hat{\lambda} \leq \frac{1}{2}$ , što daje implicitnu gornju granicu  $u \approx \frac{1}{2}$  za  $\rho$  u Algoritmu 3.2.1. S druge strane, nije poželjno previše smanjiti  $\lambda_k$  te se nameće donja granica  $l = \frac{1}{10}$ , tj. ako je dobiven  $\hat{\lambda} \leq 0.1$ , tada se uzima  $\lambda_k = \frac{1}{10}$ .



Slika 3.4: Backtracking u prvoj iteraciji primjenom modela kvadratnog polinoma

Nakon prvog odabira  $\lambda_k$ , ako  $h(\lambda_k) = f(x_k + \lambda_k p_k)$  ne zadovoljava uvjet (3.3), potreban je novi backtrack. Ponovo se može iskoristiti model jednodimenzionalnog kvadratnog polinoma, ali, budući da su sada dostupne 4 informacije o  $h(\lambda)$ :  $h(0)$ ,  $h'(0)$  i posljednje dvije vrijednosti  $h(\lambda)$ , bolje opcija je primjena kubičnog polinoma za  $h$ , koji preciznije modelira dijelove funkcije  $h$  negativne zakrivljenosti i ima jedinstven lokalni minimum.

Neka su  $\lambda_{p1}$  i  $\lambda_{p2}$  posljednje dvije vrijednosti od  $\lambda_k$ . Kub modeliran sa  $h(0)$ ,  $h'(0)$ ,  $h(\lambda_{p1})$  i  $h(\lambda_{p2})$  dan je sa

$$m_{cu}(\lambda) = a\lambda^3 + b\lambda^2 + h'(0)\lambda + h(0),$$

gdje je

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\lambda_{p1} - \lambda_{p2}} \cdot \begin{bmatrix} \frac{1}{\lambda_{p1}^2} & \frac{-1}{\lambda_{p2}^2} \\ \frac{-\lambda_{p2}}{\lambda_{p1}^2} & \frac{\lambda_{p1}}{\lambda_{p2}^2} \end{bmatrix} \begin{bmatrix} h(\lambda_{p1}) - h(0) - h'(0)\lambda_{p1} \\ h(\lambda_{p2}) - h(0) - h'(0)\lambda_{p2} \end{bmatrix}.$$

Tada je točka lokalnog minimuma  $\hat{\lambda}$  jednaka

$$\frac{-b + \sqrt{b^2 - 3ah'(0)}}{3a} \quad (3.18)$$

Kao i ranije, nameću se gornja i donja granica,  $u = \frac{1}{2}$  i  $l = \frac{1}{10}$ . Nadalje, moguće je zadržati  $\nabla f(x_k + \lambda_k p_k)$  na vrlo malom dodatnom trošku sve dok se računa samo  $f(x_k + \lambda_k p_k)$ . U tom se slučaju,  $m_c u(\lambda)$  interpolira  $h(0)$ ,  $h'(0)$ ,  $h(\lambda_{prev})$ ,  $h'(\lambda_{prev})$  u svakom backtracku.

Poželjno je u algoritam implementirati dodatna dva uvjeta zbog zaustavljanja beskonačne petlje i napuštanja područja od interesa. Prvi uvjet je definiranje duljine minimalnog koraka za testiranje konvergencije. Ukoliko uvjet (3.3) nije zadovoljen i  $\|\lambda_k p_k\|_2$  je manji od minimalnog koraka, line search staje. Obzirom da će na kraju iteracije  $x_k$  svejedno konvergirati, kriterij minimalnog koraka eliminira ulazak u beskonačnu petlju ukoliko  $p_k$  nije smjer silaska. S druge strane, definira se duljina maksimalno dopuštenog koraka zbog sprečavanja uzimanja koraka koji bi rezultirali izlaskom algoritma iz područja od interesa.

## Poglavlje 4

# Trust region model - Model regije povjerenja

U prethodnom se poglavlju razmatralo kako pronaći korak prihvatljive duljine za dani smjer silaska, pri čemu je osnovna pretpostavka bila da će smjer silaska biti kvazi-Newtonov smjer te da će pokušaj prvog koraka uvijek biti kvazi-Newtonov korak. Rezultat prethodnog poglavlja bio je backtracking algoritam ukomponiran u dane pretpostavke za pokušaj postizanja globalne konvergencije bez gubitka svojstava lokalne konvergencije kvazi – Newtonove metode. U ovom poglavlju se isto želi postići takva globalna konvergencija, ali se pritom odbacuje pretpostavka da korak mora biti u kvazi-Newtonovom smjeru silaska.

Pretpostavka da kvazi-Newtonov korak nije zadovoljavajuć, upućuje na to da model kvadratnog polinoma neadekvatno modelira funkciju  $f$  u tom području. U line search algoritmu je zadržan isti smjer koraka i odabire se kraći korak, koji je određen pomoću jednodimenzionalnog modela kvadratnog ili kubičnog polinoma, temeljenog na poznatim informacijama o funkciji i gradijentu u kvazi-Newtonovom smjeru. Iako uspješna, metoda ima nedostataka da se  $n$ -dimenzionalni model kvadratnog polinoma ne koristi više puta, uključujući Hesseovu matricu modela. Stoga se u ovom poglavlju, prije određivanja smjera koraka, prvo određuje duljina koraka.

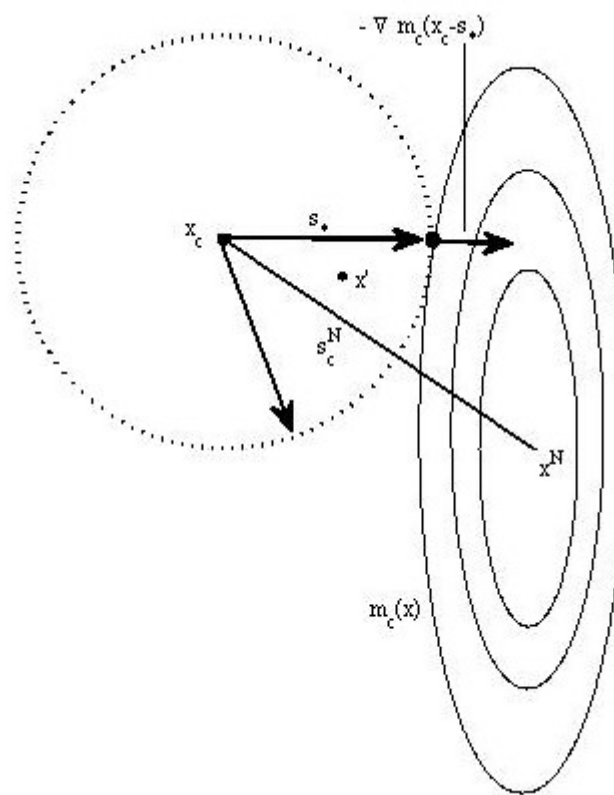
Pretpostavi li se da je poznata točka  $x_c$  i neka procjena  $\delta_c$  maksimalne duljine uspješnog koraka, postavlja se pitanje: kako najbolje odabrati korak maksimalne duljine  $\delta_c$  iz točke  $x_c$ ? Od početka se smatra da je kvazi-Newtonov korak  $s_c^N$  razuman korak, jer je to korak iz  $x_c$  prema globalnom minimumu lokalnog modela kvadratnog polinoma  $m_c$  (ako je Hesseova matrica modela,  $H_c$  pozitivno definitna). Pridoda li se ideja o ograničavanju duljine maksimalnog koraka sa  $\delta_c > 0$ , tada je odgovor na dano pitanje, da se pokuša uzeti korak

$s_c$  koji rješava

$$\min m_c(x_c + s) = f(x_c) + \nabla f(x_c)^T s + \frac{1}{2} s^T H_c s, \quad (4.1)$$

uz uvjet  $\|s\|_2 \leq \delta_c$ .

Problem (4.1) čini bazu trust region modela minimizacije. Naziv proizlazi iz promatranja  $\delta_c$  kao pružatelja *regije* u kojoj se vjeruje da  $m_c$  adekvatno modelira  $f$ . Rješenje problema dano je idućom lemom čiji se dokaz može pronaći u knjizi Dennis, Schnabel [1].



Slika 4.1: Rješenje problema (4.1)

**Lema 4.0.1.** Neka je  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  dva puta neprekidno diferencijabilna,  $H_c \in \mathbb{R}^{n \times n}$  simetrična i pozitivno definitna i neka  $\|\cdot\|$  označava  $l_2$  normu. Tada se problem (4.1) može riješiti sa

$$s(\mu) := -(H_c + \mu I)^{-1} \nabla f(x_c) \quad (4.2)$$

za jedinstven  $\mu \geq 0$  tako da je  $\|s(\mu)\| = \delta_c$ , osim ako je  $\|s(0)\| \leq \delta_c$ , gdje je u tom slučaju rješenje  $s(0) = s_c^N$ . Za bilo koji  $\mu \geq 0$ ,  $s(\mu)$  definira smjer silaska za  $f$  iz  $x_c$ .

Problem Leme 4.0.1 je u tome što u algoritmu minimizacije ne postoji konačna metoda određivanja  $\mu_c > 0$  tako da vrijedi  $\|s(\mu_c)\|_2 = \delta_c$ , kada je  $\delta_c < \|H_c^{-1}\nabla f(x_c)\|_2$ . Računske metode za aproksimativno rješenje problema (4.1) dane su u sljedeća dva potpoglavlja. Jedna metoda je lokalno ograničeni optimalni (ili “hook”) korak, koja pronalazi  $\mu_c$  takav da je  $\|s(\mu_c)\|_2 \cong \delta_c$  i uzima korak  $x_+ = x_c + s(\mu_c)$ . Druga metoda je dogleg korak, koji uzima po dijelovima linearnu aproksimaciju do krivulje  $s(\mu)$  i uzme  $x_+$  kao točku sa te aproksimacije za koju vrijedi  $\|x_+ - x_c\| = \delta_c$ . Nema garancije da će  $x_+$ , koji aproksimativno ili egzaktno rješava (4.1), biti prihvatljiva sljedeća točka, ali postoji nada da će biti prihvatljiva za dobar korak ograničenja  $\delta_c$ . Trust region algoritam ima sljedeći oblik:

#### Algoritam 4.0.2. Globalni korak trust region modela

Za dane  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\delta_c > 0$ ,  $x_c \in \mathbb{R}^n$ ,  $H_c \in \mathbb{R}^{n \times n}$  simetrična i pozitivno definitna:  
Ponavljaj:

(1)  $s_c :=$  aproksimativno rješenje za (4.1)

$$x_+ := x_c + s_c$$

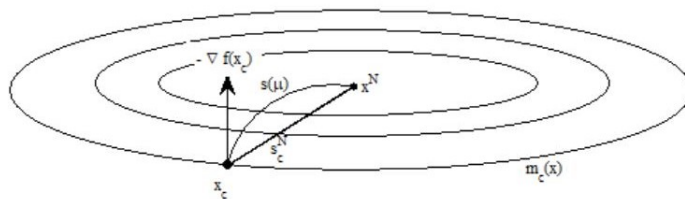
(2) odluči je li  $x_+$  prihvatljiva i izračunaj novu vrijednost  $\delta_c$   
sve dok  $x_+$  nije sljedeća prihvatljiva točka;

$$\delta_+ := \delta_c$$

Algoritam trust region modela bit će doraden u Potpoglavlju 4.3.

Dva su konačna razmatranja. Prvo, Gay [8] je pokazao da i kada  $H_c$  ima negativne svojstvene vrijednosti, rješenje za (4.1) je i dalje  $s_*$  koji zadovoljava  $(H_c + \mu I)s_* = -\nabla f(x_c)$  za neki  $\mu > 0$  takav da je  $H_c + \mu I$  najmanje pozitivno semidefinitna, što dovodi do opravdanja perturbiranja Hesseove matrice modela u pozitivno definitnu matricu  $H_c = \nabla^2 f(x_c) + \mu I$  kada  $\nabla^2 f(x_c)$  nije pozitivno definitna. Rezultirajući kvazi-Newtonov korak  $-H_c^{-1}\nabla f(x_c)$  je, u odnosu na minimum originalnog (ne pozitivno definitnog) kvadratnog modela, u nekom sfernom prostoru u okolini točke  $x_c$ . Kasnije se uzima pretpostavka da je  $H_c$  pozitivno definitna.

Drugo,  $s(\mu)$  je glatka funkcija od kvazi-Newtonog koraka  $-H_c^{-1}\nabla f(x_c)$ , kada je  $\mu = 0$ , prema  $s(\mu) \cong -(1/\mu)\nabla f(x_c)$ , kada se  $\mu$  povećava. Prema tome, kada je  $\delta_c$  vrlo malen, rješenje za (4.1) je korak duljine  $\delta_c$  u aproksimativnom smjeru najbržeg silaska. Graf 4.2 prati krivulju  $s(\mu)$ ,  $0 \leq \mu < \infty$ , istog kvadratnog modela kao Graf 4.1.

Slika 4.2: Krivulja  $s(\mu)$ 

## 4.1 Lokalno ograničen optimalni (“hook”) korak

Prva metoda izračuna aproksimativnog rješenja problema (4.1), primjenom Leme 4.0.1, kada je  $\|s(0)\|_2 > \delta_c$ , je pronalazak  $s(\mu) := -(H_c + \mu I)^{-1} \nabla f(x_c)$  tako da vrijedi  $\|s(\mu)\|_2 \cong \delta_c$  i nakon toga se provjerava je li nova točka  $x_+ := x_c + s(\mu)$  prihvatljiva ili ne. U ovom potpoglavlju promatra se algoritam pronalaska aproksimativnog rješenja  $\mu_c$  skalarne jednadžbe

$$\Phi(\mu) := \|s(\mu)\|_2 - \delta_c = 0 \quad (4.3)$$

Jedna od ideja je primjena lokalnog modela problema rješavanja. U slučaju jednodimenzionalne verzije jednadžbe (4.3),

$$\Phi(\mu) = |-(\mu + h)^{-1} g_c| - \delta_c$$

dan je lokalni model oblika

$$m_c(\mu) = \frac{\alpha_c}{\beta_c + \mu} - \delta_c$$

s dva slobodna parametara  $\alpha$  i  $\beta$ .

Prije razmatranja parametara  $\alpha$  i  $\beta$ , obzirom da se prelazi u novu poditeraciju pronalaska  $\mu$ , unutar glavne iteracije pronalaska  $x$ , bitno je razlikovati vrijednosti  $\alpha_c$ ,  $\beta_c$  i  $\mu_c$ , koje se mijenjaju unutar iteracija po  $\mu$ , od trenutnih vrijednosti  $\delta_c$ ,  $x_c$ ,  $\nabla f(x_c)$  i  $H_c$ , koje dolaze iz vanjske iteracije po  $x$  i ne mijenjaju se tijekom rješavanja unutarnje iteracije, tj. tijekom rješavanja jednadžbe (4.3). Prema tome, radi lakšeg razlikovanja, vrijednosti unutarnje iteracije će biti naznačene podebljanim fontom, a “konstante” unutarnje iteracije će biti pisane normalnim fontom. Pritom je vrijednost  $\mu_c$  jednaka posljednjoj aproksimaciji unutarnjih iteracija od  $\mu_*$  (egzaktno rješenje od (4.3)) te se  $x_c$  i  $\mu_c$  koriste za definiranje sljedećeg koraka vanjske iteracije sa  $x_+ = x_c + s(\mu_c)$ .

Model  $m_c(\mu)$  ima dva slobodna parametara  $\alpha_c$  i  $\beta_c$  te se čini razumnim da se, kao i u Newtonovoj metodi, odaberu na način da zadovoljavaju sljedeća dva uvjeta:

$$m_c(\mu_c) = \frac{\alpha_c}{\beta_c + \mu_c} - \delta_c = \Phi(\mu_c) = \|s(\mu_c)\|_2 - \delta_c$$



i

$$m_c'(\mu_c) = -\frac{\alpha_c}{(\beta_c + \mu_c)^2} = \Phi'(\mu_c) = -\frac{s(\mu_c)^T (H_c + \mu_c I)^{-1} s(\mu_c)}{\|s(\mu)\|_2}$$

Slijedi:

$$\alpha_c = -\frac{(\Phi(\mu_c) + \delta_c)^2}{\Phi'(\mu_c)}, \quad (4.4)$$

$$\beta_c = -\frac{\Phi(\mu_c) + \delta_c}{\Phi'(\mu_c)} - \mu_c. \quad (4.5)$$

 Nakon modeliranja  $m_c(\mu)$ , odabire se  $\mu_+$  tako da vrijedi  $m_c(\mu_+) = 0$ , odnosno

$$\mu_+ = \frac{\alpha_c}{\delta_c} - \beta_c.$$

Supstitucijom (4.3), (4.4) i (4.5) u gornji oblik dobije se

$$\begin{aligned} \mu_+ &= \mu_c - \frac{\Phi(\mu_c) + \delta_c}{\delta_c} \frac{\Phi(\mu_c)}{\Phi'(\mu_c)} \\ &= \mu_c - \frac{\|s(\mu_c)\|}{\delta_c} \frac{\Phi(\mu_c)}{\Phi'(\mu_c)} \end{aligned} \quad (4.6)$$

kao iteracija za rješavanje  $\Phi(\mu) = 0$ . Prema (4.6), kada je  $\mu_c < \mu_*$ , uzima se veći korak nego što daje Newtonova metoda, a kada je  $\mu_c > \mu_*$ , korak je manji od Newtonovog. Kako  $\mu_c \rightarrow \mu_*$ , (4.6) je sve bliža Newtonovoj metodi, pa slijedi da (4.6) konvergira  $q$ -kvadratno u  $\mu_*$ .

Nekoliko problema ostaje u transformiranju (4.6) u računski algoritam. Diskusija koja slijedi i sam algoritam bazirani su na Hebden [12] i Móre [13].

Prvo pitanje je odabir početne vrijednosti  $\mu$  u rješavanju (4.6). Reinsch [14] je pokazao da, ako (4.6) počinje s  $\mu = 0$ , tada  $\mu$  konvergira monotono prema  $\mu_*$ . Cilj je početi bliže rješenju, jer svaka iteracija od (4.6) uključuje rješavanje linearnog sustava. Móre-ova implementacija koristi aproksimativno rješenje  $\mu_-$  posljednje iteracije u (4.3) za generiranje inicijalne pretpostavke trenutnih (unutarnjih) iteracija. Móre-ovo pravilo je sljedeće: ako je trenutno ograničenje koraka  $\delta_c$  jednako  $p$  puta vrijednosti ograničenja koraka  $\delta_-$ , tada se kao inicijalna iteracija u (4.6) uzima  $\mu_0 = \frac{\mu_-}{p}$ .

U knjizi Dennis, Schnabel [1] dana je sljedeća metoda. Pami se trenutna pozicija u iteraciji: učinjen je  $x$ -korak  $s(\mu_-)$  iz  $x_-$  u  $x_c$ , dobiven je idući  $\delta_c$  iz  $\delta_-$  i želi se dobiti  $x_+$ . Prije no što se može izračunati  $H_c$ ,  $(H_- + \mu_- I)$  je i dalje u faktoriziranom obliku što troši samo  $O(n^2)$  za izračun

$$\Phi := \|s(\mu_-)\| - \delta_c \quad \text{i} \quad \Phi' := \frac{s(\mu_-)^T (H_- + \mu_- I)^{-1} s(\mu_-)}{\|s(\mu_-)\|}.$$

Analogno (4.6), koristi se

$$\boldsymbol{\mu}_0 = \boldsymbol{\mu}_- - \frac{\|s(\boldsymbol{\mu}_-)\|}{\delta_c} \frac{\boldsymbol{\Phi}}{\boldsymbol{\Phi}'}, \quad (4.7)$$

vrijednost  $\boldsymbol{\mu}$  dobivena iz  $\boldsymbol{\mu}_-$  prijašnjeg modela i novi trust region radijus  $\delta_c$ . Ako je  $\delta_c = \delta_-$ , tada je  $\boldsymbol{\mu}_0$  upravo vrijednost koja bi se dobila da se uzela još jedna iteracija prethodnog modela. S druge strane, ako je prethodna iteracija koristila Newtonov korak, tada se  $\boldsymbol{\mu}_0$  traži drugom tehnikom.

Drugi važan dio algoritma za rješavanje (4.3) je generiranje i određivanje donje i gornje granice za  $\boldsymbol{\mu}_+$ , tako da je  $\boldsymbol{\mu}_+ \in [\boldsymbol{l}_+, \boldsymbol{u}_+]$ . Obzirom da Newtonova iteracija na (4.3) uvijek podbaci  $\boldsymbol{\mu}_*$ , uzima se  $\boldsymbol{l}_0 = \boldsymbol{\mu}_c - \frac{\boldsymbol{\Phi}(0)}{\boldsymbol{\Phi}'(0)}$ , a zatim se računa

$$\boldsymbol{\mu}_+^N = \boldsymbol{\mu}_c - \frac{\boldsymbol{\Phi}(\boldsymbol{\mu}_c)}{\boldsymbol{\Phi}'(\boldsymbol{\mu}_c)}$$

zajedno sa svakim izračunom (4.6) i određivanjem nove donje granice  $\boldsymbol{l}_+ = \max\{\boldsymbol{l}_c, \boldsymbol{\mu}_+^N\}$ , gdje je  $\boldsymbol{l}_c$  trenutna donja granica. Kako je

$$\delta_c = \|(H_c + \boldsymbol{\mu}_* I)^{-1} \nabla f(x_c)\|_2 < \frac{\|\nabla f(x_c)\|_2}{\boldsymbol{\mu}_*}$$

jer je  $H_c$  pozitivno definitna i  $\boldsymbol{\mu}_* > 0$ , za početnu gornju granicu  $\boldsymbol{u}_0$  od  $\boldsymbol{\mu}_*$  uzima se  $\frac{\|\nabla f(x_c)\|_2}{\delta_c}$ , a u svakoj iteraciji, ako je  $\boldsymbol{\Phi}(\boldsymbol{\mu}_c) < 0$ , nova gornja granica je određena sa  $\boldsymbol{\mu}_+ = \min\{\boldsymbol{u}_c, \boldsymbol{\mu}_c\}$ , gdje je  $\boldsymbol{u}_c$  trenutna gornja granica.

Ako kod neke iteracije  $\boldsymbol{\mu}_+$  nije u skupu  $[\boldsymbol{l}_+, \boldsymbol{u}_+]$ , tada se slijedi Móre [13] u odabiru  $\boldsymbol{\mu}_+$  sa

$$\boldsymbol{\mu}_+ = \max\{(\boldsymbol{l}_+ \cdot \boldsymbol{u}_+)^{1/2}, 10^{-3} \boldsymbol{u}_+\}, \quad (4.8)$$

gdje drugi argument osigurava vrijednost  $\boldsymbol{\mu}_+$  kada je  $\boldsymbol{l}_+$  vrijednost blizu 0. (4.8) se primjenjuje za definiranje  $\boldsymbol{\mu}_0$  svaki put kada se (4.7) ne može primijeniti zbog prethodne iteracije u kojoj je korišten Newtonov korak.

Na kraju, (4.3) se ne rješava s velikom preciznošću, već se uzima

$$\|s(\boldsymbol{\mu})\|_2 \in \left[ \frac{3\delta_c}{4}, \frac{3\delta_c}{2} \right].$$

Razlog tome je što se trust region nikad ne smanjuje ili povećava faktorom manjim od 2, pa prema tome, ako je trenutni radijus  $\delta_c$ , prethodni je bio ili veći ili jednak  $2\delta_c$ , ili manji ili jednak  $\delta_c/2$ . Dakle, razmatra se da stvarna vrijednost  $\delta_c$  bude prilično proizvoljna unutar raspona  $[\frac{3\delta_c}{4}, \frac{3\delta_c}{2}]$  i čini se razumnim dopustiti da  $\|s(\boldsymbol{\mu})\|_2$  ima bilo koju vrijednost u tom rasponu. Postoje i druge implementacije, na primjer Móre [13] zahtijeva  $\|s(\boldsymbol{\mu})\|_2 \in [0.9\delta_c, 1.1\delta_c]$ , i ne zna se je li ijedna bolja od druge. U praksi je pokazano da su oba izbora zadovoljena s prosječnom vrijednosti  $\boldsymbol{\mu}$  između 1 i 2 u  $x$ -iteraciji.

**Primjer 4.1.1.** Neka je  $f(x_1, x_2) = x_1^4 + x_1^2 + x_2^2$ ,  $x_c = (1, 1, )^T$ ,  $H_c = \nabla^2 f(x_c)$ ,  $\delta_c = \frac{1}{2}$ ,  $\mu_- = 0$ . Tada je

$$\begin{aligned}\nabla f(x_c) &= (6, 2)^T, \\ H_c &= \begin{bmatrix} 14 & 0 \\ 0 & 2 \end{bmatrix} \\ s_c^N &= -H_c^{-1} \nabla f(x_c) = \left( -\frac{3}{7}, -1 \right)^T, \\ \|s_c^N\|_2 &\cong 1.088.\end{aligned}$$

Kako je  $\|s_c^N\|_2 > \frac{3}{2}\delta_c$ , Newtonov korak je predug i traži se neki  $\mu > 0$  takav da

$$\|(H_c + \mu I)^{-1} \nabla f(x_c)\|_2 \in \left[ \frac{3\delta_c}{4}, \frac{3\delta_c}{2} \right] = [0.375, 0.75]$$

(vidi Slika 4.3). Budući da je  $\mu_- = 0$ , računa se

$$\begin{aligned}l_0 &= 0 - \frac{\Phi(0)}{\Phi'(0)} \cong \frac{0.588}{0.472} \cong 1.25, \\ u_0 &= \frac{\|\nabla f(x_c)\|_2}{\delta_c} \cong \frac{6.325}{0.5} \cong 12.6, \\ \mu_0 &= (l_0 \cdot u_0)^{1/2} \cong 3.97,\end{aligned}$$

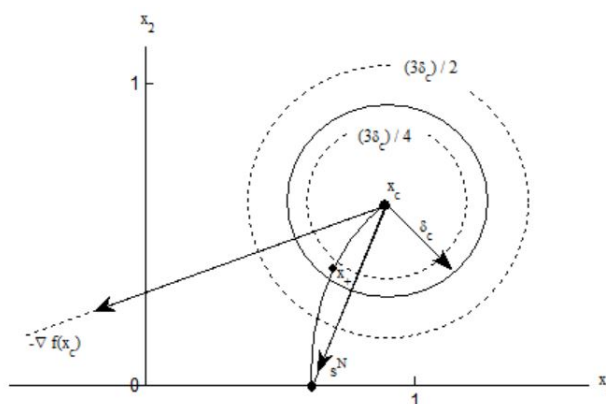
a zatim se računa  $s(\mu_0) = -(H_c + \mu_0 I)^{-1} \nabla f(x_c) \cong (-0.334, -0.335)^T$ . Kako je

$$\|s(\mu_0)\|_2 \cong 0.473 \in [0.375, 0.75],$$

uzima se  $\mu_c = \mu_0$  i  $x_+ = x_c + s(\mu_c) \cong (0.666, 0.665)^T$  kao aproksimativno rješenje od (4.1). Lako se primijeti da u ovom slučaju nije korištena iteracija (4.6). Može se provjeriti da jedna aplikacija (4.6) rezultira sa

$$\mu_1 = \mu_0 - \frac{\Phi(\mu_0)}{\Phi'(\mu_0)} \cdot \frac{\|s(\mu_0)\|_2}{\delta_c} = 3.97 - \frac{-0.0271}{-0.0528} \cdot \frac{+0.473}{+0.5} = 3.49$$

i da je  $s(\mu_1) = (-0.343, -0.365)^T$ ,  $\|s(\mu_1)\|_2 = 0.5006$ . Uz to, (4.3) je riješeno točno sa  $\mu_* \cong 3.496$ .



Slika 4.3: "Hook" korak Primjera 4.1.1

## 4.2 Dvostruki dogleg korak

Druga metoda implementacija trust region modela je modificiran trust region algoritam kojeg je predstavio Powell [15] koji također nalazi i rješenje problema (4.1). Međutim, umjesto traženja točke  $x_+ = x_c + s(\mu_c)$  na krivulji  $s(\mu)$  tako da vrijedi  $\|x_+ - x_c\|_2 \cong \delta_c$ , aproksimira se krivulja pomoću po dijelovima linearnom funkcijom povezivajući "Cauchyjevku točku"  $C.P.$ , minimum modela kvadratnog polinoma  $m_c$  u smjeru najbržeg silaska, prema Newtonovom smjeru za  $m_c$ , kao što je prikazano na Slici 4.4. Zatim se odabire točka  $x_+$  na poligolnoj liniji tako da vrijedi  $\|x_+ - x_c\|_2 = \delta_c$ , osim ako je  $\|H_c^{-1}\nabla f(x_c)\|_2 < \delta_c$ , tada je  $x_+$  Newtonova točka.

Dva su bitna svojstva dvostruke dogleg krivulje. Prvo, ako je linearna krivulja odabrana tako da se udaljenost od početne vrijednosti  $x_c$  prema Cauchy točki  $C.P.$  preko  $\hat{N}$  do  $x_+^N$  monotono povećava, tada za bilo koji  $\delta \leq \|H_c^{-1}\nabla f(x_c)\|_2$  postoji jedinstvena točka krivulje  $x_+$  takva da je  $\|x_+ - x_c\|_2 = \delta_c$ , što proces čini dobro definiranim. Drugo svojstvo je što proces čini razumnim da se vrijednost modela kvadratnog polinoma  $m_c(x_c + s)$  monotono smanjuje kako  $s$  ide duž krivulje iz  $x_c$  prema  $C.P.$  preko  $\hat{N}$  do  $x_+^N$ .

Točka  $C.P.$  dobiva se rješavanjem

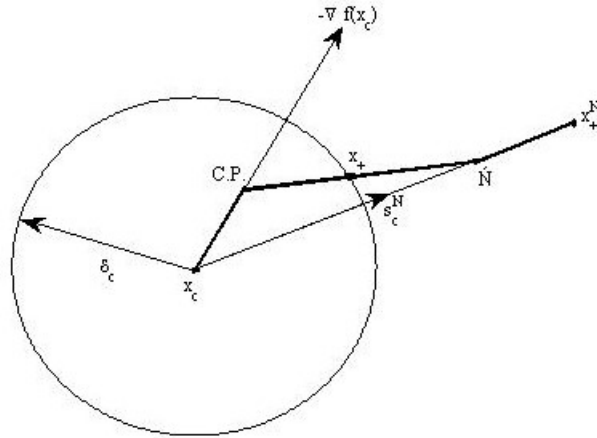
$$\min_{\lambda \in \mathbb{R}} m_c(x_c - \lambda \nabla f(x_c)) = f(x_c) - \lambda \|\nabla f(x_c)\|_2^2 + \frac{1}{2} \lambda^2 \nabla f(x_c)^T H_c \nabla f(x_c),$$

s jedinstvenim rješenjem

$$\lambda_* = \frac{\|\nabla f(x_c)\|_2^2}{\nabla f(x_c)^T H_c \nabla f(x_c)}.$$

Slijedi

$$C.P. = x_c - \lambda_* \nabla f(x_c),$$


 Slika 4.4: Dvostruka dogleg krivulja,  $x_c \rightarrow C.P. \rightarrow \hat{N} \rightarrow x_+^N$ 

i ako je

$$\delta_c \leq \lambda_* \|\nabla f(x_c)\|_2 = \frac{\|\nabla f(x_c)\|_2^3}{\nabla f(x_c)^T H_c \nabla f(x_c)},$$

algoritam uzima korak duljine  $\delta_c$  u smjeru najbržeg silaska

$$x_+ = x_c - \frac{\delta_c}{\|\nabla f(x_c)\|_2} \nabla f(x_c).$$

Kako bi dvostruka dogleg krivulja zadovoljavala prvo svojstvo, potrebno je pokazati da je Cauchyjeva točka  $C.P.$  bliža početnoj točki  $x_c$  nego Newtonova točka  $x_+^N$ . Neka su

$$s^{C.P.} = -\lambda_* \nabla f(x_c) \quad \text{i} \quad s^N = -H_c^{-1} \nabla f(x_c).$$

Tada vrijedi

$$\begin{aligned} \|s^{C.P.}\|_2 &= \frac{\|\nabla f(x_c)\|_2^3}{\nabla f(x_c)^T H_c \nabla f(x_c)} \\ &\leq \frac{\|\nabla f(x_c)\|_2^3}{\nabla f(x_c)^T H_c \nabla f(x_c)} \frac{\|\nabla f(x_c)\|_2 \|H_c^{-1} \nabla f(x_c)\|_2}{\nabla f(x_c)^T H_c^{-1} \nabla f(x_c)} \\ &= \frac{\|\nabla f(x_c)\|_2^4}{(\nabla f(x_c)^T H_c \nabla f(x_c)) (\nabla f(x_c)^T H_c^{-1} \nabla f(x_c))} \|s^N\|_2 \\ &=: \gamma \|s^N\|_2 \end{aligned} \tag{4.9}$$

Dokazuje se da je  $\gamma \leq 1$  za bilo koju pozitivno definitnu  $H_c$ , pri čemu je  $\gamma = 1$  samo ako je  $s^{C.P.} = s^N$ , pa slijedi

$$\|s^{C.P.}\|_2 \leq \gamma \|s^N\|_2 \leq \|s^N\|_2.$$

Točka na dvostrukoj dogleg krivulji je tada odabrana da vrijedi

$$\hat{N} = x_c - \eta H_c^{-1} \nabla f(x_c)$$

za neki  $\eta$  takav da zadovoljava

$$\gamma \leq \eta \leq 1$$

i da

$$m_c(x) \text{ monotono opada du\u017e linije iz C.P. prema } \hat{N}. \quad (4.10)$$

Budući da  $m_c(x)$  monotono opada od  $x_c$  prema C.P. i od  $\hat{N}$  prema  $x_+$ , (4.10) osigurava da  $m_c(x)$  monotono opada du\u017e cijele dogleg krivulje.

Kako bi (4.10) bilo zadovoljeno,  $\eta$  mora biti odabran tako da je derivacija du\u017e vektora negativna u svakoj točki du\u017e segmenta linije koja spaja C.P. i  $\hat{N}$ . Segment linije između C.P. i  $\hat{N}$  se parametrizira sa

$$x_+(\lambda) = x_c + s^{C.P.} + \lambda(\eta s^N - s^{C.P.}), \quad 0 \leq \lambda \leq 1.$$

Derivacija du\u017e vektora od  $m_c$  du\u017e linije  $x_+(\lambda)$  je tada

$$\begin{aligned} \nabla m_c(x_+(\lambda))^T (\eta s^N - s^{C.P.}) &= [\nabla f(x_c) + H_c(s^{C.P.} + \lambda(\eta s^N - s^{C.P.}))]^T (\eta s^N - s^{C.P.}) \\ &= (\nabla f(x_c) + H_c s^{C.P.})^T (\eta s^N - s^{C.P.}) + \lambda(\eta s^N - s^{C.P.})^T H_c (\eta s^N - s^{C.P.}). \end{aligned} \quad (4.11)$$

Kako je  $H_c$  pozitivno definitna, desna strana jednad\u017ebe (4.11) je monotonno rastuća funkcija od  $\lambda$ , pa je potreban zahtjev da (4.11) bude negativna za  $\lambda = 1$ , kako bi bila negativna za svaki  $0 \leq \lambda \leq 1$ . Primjenom nekih zamjena i korekcija pokazano je da je taj uvjet jednak

$$0 > (1 - \eta) (\nabla f(x_c)^T (\eta s^N - s^{C.P.})) = (1 - \eta) (\gamma - \eta) (\nabla f(x_c)^T H_c^{-1} \nabla f(x_c)),$$

što je zadovoljeno za bilo koji  $\eta \in (\gamma, 1)$ .

Dakle,  $\hat{N}$  može biti bilo koja točka u Newtonovom smjeru  $x_c + \eta s^N$  za koju je  $\eta$  između 1 i  $\gamma$  dan sa (4.9). Powellov originalni izbor je bio  $\eta = 1$ , dajući jednostruku dogleg krivulju. Međutim, računskim testiranjem pokazno je da pristranost Newtonovom smjeru pospješuje rad algoritma te su Dennis i Mei [16] predložili izbor  $\eta = 0.8\gamma + 0.2$ , rezultirajući dvostrukom dogleg krivuljom Slika 4.4.

**Primjer 4.2.1.** Neka su dani  $f(x_c)$ ,  $x_c$ ,  $H_c$  kao u Primjeru 4.1.1 i neka je  $\delta_c = 0.75$  te vrijedi

$$\nabla f(x_c) = (6, 2)^T, \quad H_c = \begin{bmatrix} 14 & 0 \\ 0 & 2 \end{bmatrix}, \quad s_c^N = \left(-\frac{3}{7}, -1\right)^T.$$

Kako je  $\|s_c^N\|_2 = 1.088 > \delta_c$ , dvostruki dogleg algoritam najprije računa korak prema Cauchyjevoj točki  $C.P.$ , koji je dan sa

$$s^{C.P.} = -\frac{40}{512} \begin{pmatrix} 6 \\ 2 \end{pmatrix} \cong \begin{pmatrix} -0.469 \\ -0.156 \end{pmatrix}.$$

Budući da je  $\|s_c^N\|_2 \cong 0.494 < \delta_c$ , algoritam nadalje računa korak do točke  $\hat{N}$  i vrijedi

$$\gamma = \frac{(40)^2}{(512)\left(\frac{32}{7}\right)} \cong 0.684,$$

$$\eta = 0.8\gamma + 0.2 \cong 0.747,$$

$$s^{\hat{N}} := \eta s_c^N \cong \begin{pmatrix} -0.320 \\ -0.747 \end{pmatrix}.$$

Kako je  $\|s^{\hat{N}}\|_2 \cong 0.813 > \delta_c$ , dvostruki dogleg korak mora biti uzduž linije koja povezuje  $C.P.$  i  $\hat{N}$ , tj.  $s_c = s^{C.P.} + \lambda(s^{\hat{N}} - s^{C.P.})$  za  $\lambda \in (0, 1)$  za koji je  $\|s_c\|_2 = \delta_c$ .

$\lambda$  se računa rješavanjem pozitivnog korijena kvadratne jednadžbe

$$\|s^{C.P.} + \lambda(s^{\hat{N}} - s^{C.P.})\|_2^2 = \delta_c^2$$

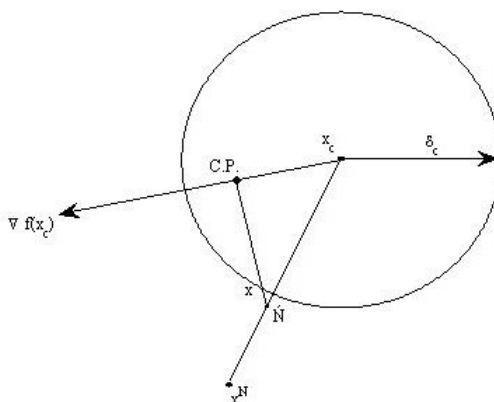
i dobije se da je  $\lambda \cong 0.867$ .

Prema tome slijedi

$$s_c = s^{C.P.} + 0.867(s^{\hat{N}} - s^{C.P.}) \cong \begin{pmatrix} -0.340 \\ -0.669 \end{pmatrix},$$

$$x_+ = x_c + s_c \cong \begin{pmatrix} 0.660 \\ 0.331 \end{pmatrix}.$$

Cijeli izračun je prikazan na Slici 4.5.



Slika 4.5: Dvostruka dogleg krivulja Primjera 4.2.1

### 4.3 Modifikacija trust region algoritma

Za dovršavanje Algoritma 4.0.2 potrebna je odluka je li točka  $x_+$ , dobivena lokalno ograničenim optimalnim (“hook”) korakom ili korakom dvostruke dogleg krivulje, zadovoljavajuća sljedeća iteracija. U slučaju kada  $x_+$  nije prihvatljiv, potrebno je smanjiti veličinu regije povjerenja (trust region), tj. smanjiti  $\delta_c$ , i minimizirati odgovarajuće modele kvadratnog polinoma. Ako je  $x_+$  zadovoljavajuć, tada se donosi odluka o tome je li trust region poželjno smanjiti, povećati ili ne mijenjati za idući korak Algoritma 4.0.2.

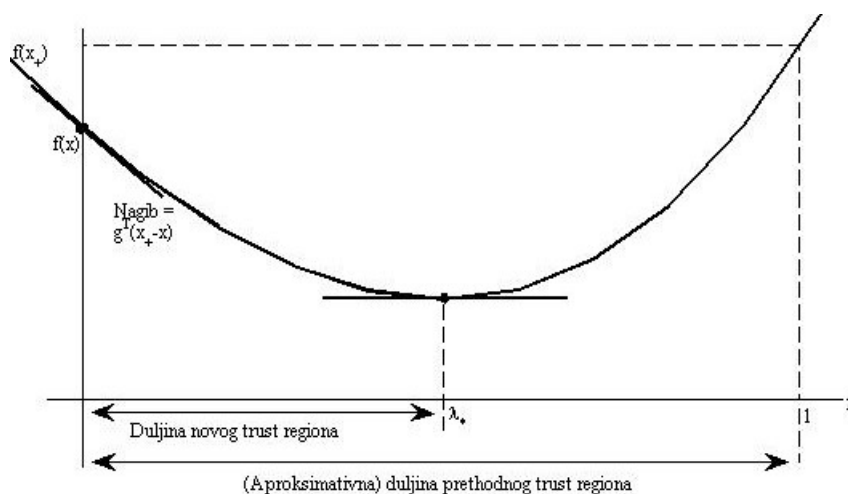
Uvjet prihvaćanja točke  $x_+$  je 1. Wolfe-ov uvjet iz Poglavlja 3 koji zadovoljava Armijo pravilo

$$f(x_+) \leq f(x_c) + \alpha g_c^T(x_+ - x_c), \quad (4.12)$$

gdje je  $g_c = \nabla f(x_c)$  ili njena aproksimacija, a  $\alpha$  je konstanta na  $(0, \frac{1}{2})$ . Ponovo se uzima  $\alpha = 10^{-4}$  tako da je (4.12) samo malo manje stroža od  $f(x_+) < f(x_c)$ . Sada ako  $x_+$  ne zadovoljava (4.12), trust region se reducira faktorom između  $\frac{1}{10}$  i  $\frac{1}{2}$  i ponovo se računa aproksimativno rješenje lokalno ograničenog minimizacijskog problema jednom od gornjih metoda. Faktor redukcije određuje se kvadratnom backtracking metodom kao i kod metode line search Algoritma 3.2.1.  $f(x_c + \lambda(x_+ - x_c))$  se modelira kvadratnim polinomom  $m_q(\lambda)$  pomoću  $f(x_c)$ ,  $f(x_+)$  i derivacije duž vektora  $g_c^T(x_+ - x_c)$  funkcije  $f$  u točki  $x_c$  u smjeru  $x_+ - x_c$ . Proširenje novog trust region radijusa  $\delta_+$  je tada moguće do minimuma gornjeg modela i postiže se za

$$\lambda_* = \frac{-g_c^T(x_+ - x_c)}{2[f(x_+) - f(x_c) - g_c^T(x_+ - x_c)]}.$$




 Slika 4.6: Reduciranje trust regiona za prihvatljiv  $x_+$ 

Slijedi,  $\delta_+ = \lambda_* \|x_+ - x_c\|_2$ . Kao i u line search algoritmu, ako

$$\lambda_* \|x_+ - x_c\|_2 \notin \left[ \frac{1}{10} \delta_c, \frac{1}{2} \delta_c \right],$$

tada se  $\delta_+$  postavlja na bližu vrijednost krajnjoj točki gornjeg intervala, tj. ako je  $\lambda_* \|x_+ - x_c\|_2 < \frac{1}{10} \delta_c$ , tada se uzima  $\delta_c = \frac{1}{10} \delta_c$  i obratno, ako je veće od  $\frac{1}{2} \delta_c$ , tada se uzima  $\delta_c = \frac{1}{2} \delta_c$ . Nadalje, može se primijetiti da je  $\|x_+ - x_c\|_2 = \delta_c$  kada se primijeni dvostruka dogleg metoda, a kada se primijeni lokalno ograničena optimalna metoda, poznata je samo informacija  $\|x_+ - x_c\|_2 \in \left[ \frac{3}{4} \delta_c, \frac{3}{2} \delta_c \right]$ .

Pretpostavi li se da je dobivena iteracija  $x_+$  koja zadovoljava (4.12), postavlja se pitanje je li  $x_+ - x_c$  puni Newtonov korak. Ako jest,  $x_+$  se uzima kao nova iteracija i izračuna se  $\delta_+$  iz novog modela te se prelazi na sljedeću iteraciju. Ako nije, tada se pokušava naći veći korak iz  $x_c$ , koristeći trenutni model.

Iako nije nužan, gornjim postupkom se može izbjeći računanje gradijenta (i Hesseove matrice) u  $x_+$ , što često predstavlja veći trošak problema. Moguće je da se trust region smanjio tijekom algoritma jer funkcija nije bila dobro aproksimirana kvadratnim polinomom na nekom području, a kad izađemo iz tog područja, gdje se funkcija ponaša puno pristojnije, trust region je poželjno ponovo proširiti.

Zanimljiva situacija kada se trust region sužava, a zatim ponovo proširuje je kada algoritam prolazi blizu točke koja djeluje kao minimum. U tom slučaju se Newtonov korak skraćuje, algoritam uzima skraćeni Newtonov korak i ponaša se kao da konvergira. Zatim algoritam pronalazi put van, pri čemu se Newtonov korak proširuje i algoritam nastav-

lja dalje. Ovo ponašanje je poželjno, jer perturbirani problem može doista imati stvaran minimum u točki distrakcije, pa je u tom slučaju poželjno brzo povećanje  $\delta_c$ .

Gornji postupak povećanja koraka provodi se na sljedeći način. Nakon što se dobi zadovoljavajući  $x_+$ , uspoređuje se stvarna redukcija funkcije  $f$ ,  $\Delta f := f(x_+) - f(x_c)$ , sa aproksimativnom (predviđenom) redukcijom,  $\Delta f_{pred} := m_c(x_+) - f(x_c)$ . Tada se  $x_+$  prihvaća kao sljedeća iteracija, osim ako ove obje redukcije nisu dovoljno dobre, tj. ako je  $|\Delta f_{pred} - \Delta f| \leq 0.1|\Delta f|$ , da postoji sumnja kako je  $\delta_c$  podcijenjen trust region radijus unutar kojeg  $m_c$  dobro aproksimira funkciju  $f$ , ili ako je stvarna redukcija funkcije  $f$  toliko velika da je implicirana prisutnost negativne zakrivljenosti, a time i neprekidno ubrzano smanjenje vrijednosti funkcije  $f$ ,  $f(x)$ , npr. kada je  $f(x_+) \leq f(x_c) + \nabla f(x_c)^T(x_+ - x_c)$ . U oba slučaja se sprema vrijednost  $x_+$  i  $f(x_+)$ , ali umjesto nastavljanja prema  $x_+$ , najprije se duplicira  $\delta_c$  i računa se novi  $x_+$  primjenom trenutnog modela. Ako uvjet (4.12) nije zadovoljen za novi  $x_+$ , odbacuje se i vraća se na zadnji dobar izračunati korak. Ako pak zadovoljava (4.12), ponavlja se postupak povećanja koraka. U praksi se na opisan način može uštedjeti velik broj poziva gradijenta.

Naposljetku, pretpostavi li se da je zadovoljavajući  $x_+$  odabran kao sljedeća iteracija, preostaje izračun novog  $\delta_+$  iz  $\delta_c$ . Dozvoljene su tri alternative: dupliciranje, raspolavljanje ili zadržavanje trenutne vrijednosti  $\delta_c$ . Uvjeti odlučivanja su donekle proizvoljni, ali važno je da ukoliko trenutni kvadratni model dobro aproksimira funkciju  $f$ , tada se trust region povećava,  $\delta_c = 2\delta_c$ , a ako vrlo loše aproksimira, tada se trust region smanjuje,  $\delta_+ = \delta_c/2$ . Npr.

$$\delta_+ = \begin{cases} \delta_c/2, & \Delta f > 0.1\Delta f_{pred} \\ 2\delta_c, & \Delta f \leq 0.75\Delta f_{pred} \\ \delta_c, & \text{inače.} \end{cases}$$

**Primjer 4.3.1.** Neka su dani  $f(x_c)$ ,  $x_c$ ,  $H_c$  kao u Primjeru 4.1.1 i pretpostavka je da je upravo učinjen korak određen u tom primjeru,

$$x_+ = x_c + s_c = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.334 \\ 0.335 \end{pmatrix} = \begin{pmatrix} 0.666 \\ 0.665 \end{pmatrix}.$$

Poznato je da vrijedi

$$\nabla f(x_c) = (6, 2)^T, \quad H_c = \nabla^2 f(x_c) = \begin{bmatrix} 14 & 0 \\ 0 & 2 \end{bmatrix}.$$

Želi se odlučiti je li  $x_+$  zadovoljavajuća točka i treba li ažurirati regiju povjerenja. Prvo se računa

$$f(x_+) = 1.083,$$

$$f(x_c) + \alpha \nabla f(x_c)^T (x_+ - x_c) = 3 - 10^{-4}(2.673) = 2.9997,$$

i slijedi da je  $x_+$  prihvatljiv. Nadalje, odlučuje se hoće li se pokušati naći veći korak u trenutnoj iteraciji računajući

$$\Delta f = f(x_+) - f(x_c) = -1.917$$

i

$$\Delta f_{pred} = m_c(x_+) - f(x_c) = \nabla f(x_c)^T s_c + \frac{1}{2} s_c^T H_c s_c = -2.673 + 0.892 = -1.781.$$

Budući da je  $\frac{|\Delta f - \Delta f_{pred}|}{|\Delta f|} = 0.071 < 0.1$ ,  $\delta_c$  se duplicira,  $\delta_+ = 2\delta_c$ , i vraća se na lokalno ograničeni optimalni korak u Algoritmu 4.0.2. Također, može se potvrditi da kada je novi radijus  $\delta_c = 1$ , Algoritam 4.0.2 odabire Newtonov korak.

Naposljetku, razmatra se kako je dobivena inicijalna procjena trust region radijusa, odnosno korak ograničenja. Ponekad korisnik može opskrbiti razumnu procjenu baziranu na svom znanju problema. Ako ne, Powell [15] predlaže korištenje duljine Cauchyjevog koraka kao inicijalni trust region radijus.

Modifikacija trust region algoritma u praksi omogućava oporavak osnovnog algoritma od loše početne vrijednosti  $\delta$ , ali uz neki trošak dodatnih iteracija, pa je inicijalni trust region vrlo bitan.

## Poglavlje 5

# Globalne metode za rješavanje sustava nelinearnih jednažbi

Nakon razmatranja globalnih metoda u Poglavljima 3 i 4, preostaje pokazati kako se Newtonova metoda za rješavanje problema nelinearnih jednažbi, danog sa

$$\begin{aligned} &\text{za danu } F : \mathbb{R}^n \longrightarrow \mathbb{R}^n, \\ &\text{nađi } x_* \in \mathbb{R}^n \text{ takav da } F(x_*) = 0, \end{aligned} \tag{5.1}$$

kombinira s globalnom metodom bezuvjetne optimizacije u globalnu metodu za rješavanje problema (5.1).

Newtonov korak za (5.1) je

$$x_+ = x_c - J(x_c)^{-1}F(x_c), \tag{5.2}$$

gdje je  $J(x_c)$  Jacobijeva matrica od  $F$  u  $x_c$ , za koji je ranije pokazano da je lokalno  $q$ -kvadratno konvergentan prema  $x_c$ , ali ne nužno i globalno konvergentan. Pretpostavimo da  $x_c$  nije blizu ni jednom rješenju  $x_*$  od (5.1). Pitanje je kako odlučiti prihvaća li se  $x_+$  kao sljedeći iteracija? Odgovor je da  $\|F(x_+)\|$  bude manje od  $\|F(x_c)\|$  za neku normu  $\|\cdot\|$ . Kao što je već spomenuto u Poglavlju 2.2, dobar izbor norme je  $l_2$  norma,  $\|F(x)\|_2^2 = F(x)^T F(x)$ , a problem (5.1) tada korespondira minimizacijskom problemu

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} F(x)^T F(x), \tag{5.3}$$

Dakle, svako rješenje za (5.1) je i rješenje za (5.3), ali mogu postojati lokalni minimumi u (5.3), koji nisu rješenja za (5.1). Iako bi za rješavanje problema (5.1) mogli koristiti minimizacijske algoritme na (5.3) bolje je rješavati originalni problem (5.1), posebno u izračunu Newtonovog koraka (5.2). Međutim, ovdje će globalne metode za (5.1) biti bazirane na danim globalnim metodama odgovarajućeg minimizacijskog problema (5.3).

Važno pitanje je sada: Koji je smjer silaska problema (5.3)? Smjer silaska je bilo koji smjer  $p$  za koji je  $\nabla f(x_c)^T p < 0$ , gdje je

$$\nabla f(x_c) = \frac{d}{dx} \sum_{i=1}^n \frac{1}{2} (f_i(x_c))^2 = \sum_{i=1}^n \nabla f_i(x_c) \cdot f_i(x_c) = J(x_c)^T F(x_c),$$

pa je najbrži smjer silaska za (5.3) smjer duž  $-J(x_c)^T F(x_c)$ . Nadalje, Newtonov smjer duž  $s^N = -J(x_c)^{-1} F(x_c)$  je smjer silaska, budući da je

$$\nabla f(x_c)^T s^N = -F(x_c)^T J(x_c) J(x_c)^{-1} F(x_c) = -F(x_c)^T F(x_c) < 0$$

sve dok je  $F(x_c) \neq 0$ . Ovo može djelovati iznenađujuće, ali je geometrijski razumno. Budući da Newtonov korak daje korijen od

$$M_c(x_c + s) = F(x_c) + J(x_c)s,$$

isto vrijedi i za minimum kvadratne funkcije

$$\begin{aligned} \hat{m}_c(x_c + s) &:= \frac{1}{2} M_c(x_c + s)^T M_c(x_c + s) \\ &= \frac{1}{2} F(x_c)^T F(x_c) + (J(x_c)^T F(x_c))^T s + \frac{1}{2} s^T (J(x_c)^T J(x_c)) s, \end{aligned} \quad (5.4)$$

jer je  $\hat{m}_c(x_c + s) \geq 0$  za sve  $s$  i  $\hat{m}_c(x_c + s^N) = 0$ . Prema tome,  $s^N$  je smjer silaska za  $\hat{m}_c$ , a kako su gradijenti u  $x_c$  od  $\hat{m}_c$  i  $f$  jednaki, isto vrijedi i za smjer silaska funkcije  $f$ .

Time je motiviran razvoj globalnih metoda za (5.1), a bazire se na primjeni algoritama iz Poglavlja 3 i 4 za model kvadratnog polinoma  $\hat{m}_c(x)$  u (5.4). Sve dok je  $J(x_c)$  regularna matrica i kako je  $\nabla \hat{m}_c^2(x_c) = J(x_c)^T J(x_c)$ , model je pozitivno definitan što je povezano s činjenicom da je  $x_c + s^N$  jedinstven korijen od  $M_c(x)$ , a tada i jedinstven minimum za  $\hat{m}_c(x)$ . Minimum polinoma  $\hat{m}_c(x)$  je Newtonova točka originalnog problema i njegovi smjerovi silaska su smjerovi silaska funkcije  $f$ , jer je  $\nabla \hat{m}_c(x_c) = \nabla f(x_c)$ . Prema tome, metode bazirane na danom modelu, minimizirajući  $\hat{m}_c(x)$ , će kombinirati Newtonovu metodu nelinearnih jednadžbi s nekom od globalnih metoda pridruženog minimizacijskog problema.

Sada se direktno mogu primijeniti metode Poglavlja 3 i 4 na problem nelinearnih jednadžbi. Sve dok je  $J(x_c)$  dovoljno dobro uvjetovana,  $J(x_c)^T J(x_c)$  je sigurno pozitivno definitna i dani algoritmi se primjenjuju uz definiranu minimizacijsku funkciju  $f(x) = \frac{1}{2} \|F(x)\|_2^2$ , Newtonov korak  $-J(x)^{-1} F(x)$  i pozitivno definitni kvadratni model (5.4). Dakle, u line search algoritmu se u Newtonovom smjeru traži dovoljni pad funkcije  $\|F(x)\|_2$ , a u trust region algoritmu se aproksimativno minimizira  $\hat{m}_c(x_c + s)$  uz uvjet  $\|s\|_2 \leq \delta_c$ . Ako je  $\delta_c \geq \|J(x_c)^{-1} F(x_c)\|_2$ , tada je korak pokušaja Newtonov korak, a inače je, za lokalno ograničeni optimalni korak,

$$s = -(J(x_c)^T J(x_c) + \mu_c I)^{-1} J(x_c)^T F(x_c) \quad (5.5)$$

za  $\mu_c$  takav da je  $\|s\|_2 \cong \delta_c$ .

Primijenom bilo koje globalne metode, očekuje se da će se u nekom trenutku primijeniti Newtonov korak za  $F(x) = 0$ . Idući primjer prikazuje princip rada line search metode u točki kada Newtonov korak nije zadovoljavajuć.

**Primjer 5.0.1.** Neka je  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ,

$$F(x) = \begin{bmatrix} x_1^2 + x_2^2 - 2 \\ e^{x_1-1} + x_2^3 - 2 \end{bmatrix},$$

sa korijenom  $x_+ = (1, 1)^T$ , i neka je  $x_0 = (2, 0.5)^T$ . Definira se  $f(x) = \frac{1}{2}F(x)^T F(x)$ . Tada je

$$J(x_0) = \begin{bmatrix} 4 & 1 \\ e & 0.75 \end{bmatrix}, \quad F(x_0) \cong \begin{bmatrix} 2.25 \\ 0.843 \end{bmatrix},$$

i

$$s_0^N = -J(x_0)^{-1} f(x_0) \cong \begin{bmatrix} -3.00 \\ 9.74 \end{bmatrix}.$$

Algoritam line search (ili pretraživanje po pravcu) će izračunati  $x_+ = x_0 + \lambda_0 s_0^N$  počevši sa  $\lambda_0 = 1$  i smanjujući  $\lambda_0$ , ako je potrebno, sve dok je  $f(x_+) < f(x_0) + 10^{-4} \lambda_0 \nabla f(x_0)^T s_0^N$ . Za  $\lambda_0 = 1$ ,

$$x_+ = x_0 + s_0^N \cong \begin{bmatrix} -1.00 \\ 10.24 \end{bmatrix}, \quad F(x_+) \cong \begin{bmatrix} 104 \\ 1071 \end{bmatrix},$$

tako da je Newtonov korak očito nezadovoljavajuć. Prema tome  $\lambda_0$  se reducira kvadratnim backtrackom, računajući,

$$\lambda_1 = \frac{-\nabla f(x_0)^T s_0^N}{2[f(x_+) - f(x_0) - \nabla f(x_0)^T s_0^N]}. \quad (5.6)$$

U ovom slučaju,  $f(x_+) \cong 5.79 \times 10^5$ ,  $f(x_0) \cong 2.89$ ,  $\nabla f(x_0)^T s_0^N = -F(x_0)^T F(x_0) \cong -5.77$ , tako da je (5.6) dano sa  $\lambda_1 \cong 4.99 \times 10^{-6}$ . Kako je  $\lambda_1 < 0.1$ , line search algoritam postavlja  $\lambda_1 = 0.1$ ,

$$x_+ = x_0 + 0.1 s_0^N \cong \begin{bmatrix} 1.70 \\ 1.47 \end{bmatrix}, \quad F(x_+) \cong \begin{bmatrix} 3.06 \\ 3.21 \end{bmatrix}.$$

Ovo je i dalje nezadovoljavajuće pa se primijenjuje kubičan backtrack koji rezultira sa  $\lambda_2 \cong 0.0659$ . Kako je  $\lambda_2 > \frac{1}{2} \lambda_1$ , algoritam postavlja  $\lambda_2 = \frac{1}{2} \lambda_1 = 0.05$ ,

$$x_+ = x_0 + 0.05 s_0^N \cong \begin{bmatrix} 1.85 \\ 0.987 \end{bmatrix}, \quad F(x_+) \cong \begin{bmatrix} 2.40 \\ 1.30 \end{bmatrix}.$$

Ova točka je i dalje nezadovoljavajuća, jer je  $f(x_+) \cong 3.71 > f(x_0)$ , algoritam ponovno provodi kubični backtrack rezultirajući sa  $\lambda_3 \cong 0.0116$  koji se nalazi unutar intervala  $[\lambda_2/10, \lambda_2/2] = [0.005, 0.025]$ . Sada je

$$x_+ = x_0 + 0.0116s_0^N \cong \begin{bmatrix} 1.965 \\ 0.613 \end{bmatrix}, \quad F(x_+) \cong \begin{bmatrix} 2.238 \\ 0.856 \end{bmatrix}.$$

Dobivena točka je zadovoljavajuća, budući da je

$$f(x_+) \cong 2.87 < 2.89 \cong f(x_0) + 10^{-4}(0.0116)\nabla f(x_0)^T s_0^N,$$

pa se uzima  $x_1 = x_+$  i nastavlja se sa sljedećom iteracijom.

Kod sljedeće iteracije, dobiveni su

$$s_1^N = -J(x_1)^{-1}F(x_1) \cong \begin{bmatrix} -1.22 \\ 2.07 \end{bmatrix},$$

$$x_2^N = x_1 + s_1^N \cong \begin{bmatrix} 0.750 \\ 2.68 \end{bmatrix}, \quad F(x_2^N) \cong \begin{bmatrix} 5.77 \\ 18.13 \end{bmatrix}$$

što je ponovo nezadovoljavajuće. Međutim, prvi backtrack korak je uspješan: line search algoritam računa  $\lambda_1 = 0.0156$  i kako je to manje od 0.1, postavlja  $\lambda_1 = 0.1$ ,

$$x_+ = x_1 + s_1^N \cong \begin{bmatrix} 1.84 \\ 0.820 \end{bmatrix}, \quad F(x_+) \cong \begin{bmatrix} 2.07 \\ 0.876 \end{bmatrix}.$$

Korak se prihvaća, budući da vrijedi

$$f(x_+) \cong 2.53 < 2.87 \cong f(x_1) + 10^{-4}(0.1)\nabla f(x_1)^T s_1^N.$$

U sljedećoj iteraciji, dobiveni su

$$s_2^N = -J(x_2)^{-1}F(x_2) \cong \begin{bmatrix} -0.0756 \\ 0.0437 \end{bmatrix},$$

$$x_3^N = x_2 + s_2^N \cong \begin{bmatrix} 1.088 \\ 1.257 \end{bmatrix}, \quad F(x_3^N) \cong \begin{bmatrix} 0.762 \\ 1.077 \end{bmatrix},$$

pa je Newtonov korak vrlo dobar. Odavde nadalje, Newtonova metoda konvergira  $q$ -kvadratno prema  $x_* = (1, 1)^T$ .

Problem se javlja u slučaju kada je  $J$  gotovo singularna u trenutnoj iteraciji  $x_c$ , pri čemu se ne može izračunati Newtonov smjer  $s^N = -J(x_c)^{-1}F(x_c)$  i Hesseova matrica modela  $J(x_c)^T J(x_c)$  je također singularna. Kako bi se uočio takav slučaj, provodi se QR faktorizacija<sup>5</sup> na  $J(x_c)$  i ako je  $R$  regularna, procijenjuje se uvjetovanost.

S druge strane, ako je  $R$  singularna ili je procijenjen broj uvjetovanosti matrice veći od  $macheps^{-1/2}$ , tada se kvadratni model perturbira u

$$\hat{m}_c(x_c + s) = \frac{1}{2}F(x_c)^T f(x_c) + (J(x_c)^T F(x_c))^T s + \frac{1}{2}s^T H_c s,$$

gdje je

$$H_c = J(x_c)^T J(x_c) + (n \cdot macheps)^{1/2} \|J(x_c)^T J(x_c)\|_1 \cdot I.$$

Broj uvjetovanosti  $H_c$  je otprilike oko  $macheps^{-1/2}$ <sup>6</sup>.

Iz Poglavlja 4 se zna da Newtonov korak minimuma gornjeg modela,  $s^N = -H_c^{-1}J(x_c)^T F(x_c)$ , rješava problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \hat{m}_c(x_c + s) &= \|J(x_c)s + F(x_c)\|_2^2 \\ \text{uz uvjet} \quad &\|s\|_2 \leq \delta \end{aligned}$$

za neki  $\delta > 0$  i da je smjer silaska za  $f(x) = \frac{1}{2}\|F(x_c)\|_2^2$ .

---

<sup>5</sup>QR faktorizacija,  $A = QRP$ , gdje je  $Q$  ortogonalna matrica,  $R$  gornjetrokutasta, a  $P$  matrica permutacije, je transformacija matrice  $A$  u  $R$  množenjem zdesna matrice  $A$  s  $n - 1$  ortogonalnih matrica  $Q_i$ , gdje su  $Q_i$  Householderove transformacije oblika  $Q_i = I - u_i u_i^T$ , za  $(u_i)_j = 0, j = 1, \dots, i - 1$  i ostalim elementima  $u_i$  izabranima tako da je  $Q_i$  ortogonalna i inducira nule na željenim stupcima.

<sup>6</sup>Algoritam 1.0.13



# Poglavlje 6

## Zaustavljanje, skaliranje i testiranje

Prilikom rješavanja problema nelinearnih jednadžbi i minimizacijskog problema, postoje tri periferna problema osnovnih matematičkih razmatranja:

- problem skaliranja - prilagodba loše skaliranih problema, u smislu da su ovisne i neovisne varijable različitih veličina
- problem zaustavljanja - kako odrediti kada zaustaviti algoritam u konačnoj aritmetici
- testiranje i usporedba algoritama

### 6.1 Skaliranje

U rješavanju mnogih problema “stvarnog svijeta” neke ovisne i neovisne varijable mogu uvelike varirati u veličini. Raspon veličina varijabli nazivamo *skalom* varijable i promatramo učinak široko raspršenih skala u našim algoritmima. Skaliranje utječe na izračune poput  $\|x_+ - x_c\|_2$ . Primjer lošeg skaliranja je minimizacijski problem s dvije varijable od kojih je prva neovisna varijabla u rasponu  $[10^3, 10^4]$  metara, a druga u rasponu  $[10^{-6}, 10^{-5}]$  sekundi. U tom slučaju izračun ignorira drugu varijablu (vrijeme) koju je tada poželjno skalirati, odnosno promijeniti veličnu mjerne jedinice. Primjerice, promijeni li se prva varijabla u kilometre, a druga u mikrosekunde, tada će obje varijable biti u rasponu  $[1, 10]$  i problem skaliranja u izračunu  $\|x_+ - x_c\|_2$  će biti riješen. Promjena neovisne varijable zapravo odgovara promjeni  $\hat{x} = D_x x$ , gdje je  $D_x$  dijagonalna matrica skaliranja

$$D_x = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^6 \end{bmatrix}. \quad (6.1)$$

Nameće se prirodno pitanje utjecaja skaliranja varijabli na rezultirajući korak u odnosu na korak dobiven istom globalnom metodom u inicijalnom prostoru varijabli. Odgovor

je da skaliranje nema utjecaj na Newtonov korak, ali dovodi do razlike u smjeru najbržeg silaska, što opet nema učinak kod line search algoritma u Newtonovom smjeru, ali rezultira promjenom u koraku trust region algoritma.

Promotrimo minimizacijski problem transformacije varijable  $x$  u  $\hat{x} = Tx$ ,  $\hat{f}(\hat{x}) = f(T^{-1}\hat{x})$ , gdje je  $T \in \mathbb{R}^{n \times n}$  regularna matrica. Pokaže se da vrijedi

$$\nabla \hat{f}(\hat{x}) = (T^{-1})^T \nabla f(x), \quad \nabla^2 \hat{f}(\hat{x}) = (T^{-1})^T \nabla^2 f(x) T^{-1},$$

pa su Newtonov korak i smjer najbržeg silaska u prostoru novih varijabli dani sa

$$\hat{s}^N = -((T^{-1})^T \nabla^2 f(x) T^{-1})^{-1} ((T^{-1})^T \Delta f(x)) = -T \nabla^2 f(x)^{-1} \nabla f(x),$$

$$\hat{s}^{SD} = -(T^{-1})^T \nabla f(x),$$

ili u prostoru starih varijabli,

$$s^N = T^{-1} \hat{s}^N = -\nabla^2 f(x)^{-1} \nabla f(x),$$

$$s^{SD} = T^{-1} \hat{s}^{SD} = -T^{-1} (T^{-1})^T \nabla f(x) = -(T^T T)^{-1} \nabla f(x).$$

Rezultat nije iznenađujuć, obzirom da Newtonov korak ide prema najnižoj točki kvadratnog modela, na koju ne utječe promjena veličine varijable  $x$ . S druge strane, smjer najbržeg silaska ovisi o tome što se smatra veličinom koraka u svakom smjeru te ima smisla ako je korak u veličini jedne varijable smjera  $x_i$  relativno jednake duljine kao korak u veličini druge varijable smjer  $x_j$ .

Prema tome, preferirano rješenje problema skaliranja je da korisnik odabere veličinu prostora varijabli tako da svaka komponenta od  $x$  bude približno iste dimenzije, što nije uvijek moguće. U tu svrhu se primjenjuje transformacija prostora varijabli pomoću dijagonalne matrice skaliranja  $D_x$ . U algoritmima se ona postavlja da odgovara željenoj promjeni veličine varijabli, ali algoritmi su i dalje pisani u inicijalnom prostoru varijabli, pa izraz  $\|x_+ - x_c\|_2$  postaje  $\|D_x(x_+ - x_c)\|_2$ , a korak metode najbržeg silaska i “hook” korak su tada jednaki

$$x_+ := x_c - \lambda D_x^{-2} \nabla f(x_c),$$

$$s(\mu) := -(H_c + \mu D_x^2)^{-1} \nabla f(x_c),$$

respektivno, dok Newtonov smjer ostaje nepromijenjen.

Pozitivno dijagonalna matrica skaliranja  $D_x$  je specificirana od strane korisnika, osiguravajući  $n$  vrijednosti  $\text{typ}x_i$ ,  $i = 1, \dots, n$ , zadajući veličinu svake varijable  $x_i$ . Algoritam postavlja  $(D)_{ii} = (\text{typ}x_i)^{-1}$ , rezultirajući veličinom svake transformirane varijable  $\hat{x}_i = (D_x)_{ii} x_i$  oko 1. Na primjer, postavite li se veličine varijabli  $\text{typ}x_1 = 10^3$  i  $\text{typ}x_2 = 10^{-6}$ , tada je  $D_x$  jednaka (6.1). Ukoliko za  $x_i$  nije potrebno skaliranje,  $\text{typ}x_i$  se postavlja na 1.

Opisana metoda skaliranja nije uvijek uspješna i ponekad je potrebno dinamičko skaliranje jer  $x_i$  varira zbog velikog broja redova veličina, ali ono nije uključeno u naše algoritme.

Nadalje, nužno je razmotriti i skaliranje ovisnih varijabli. U problemima minimizacije, skaliranje funkcije  $f$  bitno je samo za kriterij zaustavljanja. U ostalim izračunima, ta promjena nema nikakvih posljedica.

S druge strane, u rješavanju sustava nelinearnih jednadžbi, razlike u veličini komponentnih funkcija  $f_i$  mogu rezultirati jednakim problemima kao razlike u veličini neovisnih komponenata. Newtonov korak je neovisan o skaliranju, ali globalne metode za nelinearne jednadžbe zahtijevaju smanjenje u  $\|F\|_2$  i jasno je da, ukoliko su veličine dviju komponentnih funkcija od  $F(x)$  vrlo različite, tada će manja komponentna funkcija biti virtualno ignorirana.

Iz tog razloga se u algoritmima koristi pozitivna dijagonalna matrica skaliranja  $D_F$  za ovisne varijable  $F(x)$ , koja radi po istom principu kao  $D_x$  za  $x$ . Dijagonalna matrica  $D_F$  je odabrana tako da sve komponente od  $D_F F(x)$  budu otprilike jednake veličine u točkama koje nisu previše blizu korijenu i koristi se za skaliranje  $F$  u svakom modelu nelinearnih jednadžbi. Afini model postaje  $D_F M_c$ , a model kvadratnog polinoma funkcija globalnog koraka postaje  $\hat{m}_c = \frac{1}{2} \|D_F M_c\|_2^2$ . Dani algoritmi su implementirani tako da korisnik treba samo specificirati inicijalnu  $D_F$ , unosom vrijednosti  $\text{typ} f_i$ ,  $i = 1, \dots, n$ , uobičajenih veličina svake  $f_i$  u točkama koje nisu previše blizu korijenu. Algoritam tada postavlja  $(D_F)_{ii} = \text{typ} f_i^{-1}$ , odnosno sprema kao varijablu  $S_F \in R^n$ ,  $(S_F)_i = (D_F)_{ii}$ .

## 6.2 Kriterij zaustavljanja

Jedan od važnijih čimbenika je pitanje kada algoritam prestaje. Kada je poželjno algoritam zaustaviti? Kada je problem riješen? Kada algoritam dolazi do zastoja i kada ne daje rezultat te je li ušao u bekonačnu petlju bez izlaska?

Promotrimo najprije kriterij zaustavljanja bezuvjetne optimizacije, pri čemu je najvažnije pitanje: Je li problem riješen? Nužan i dovoljan uvjet da  $x$  bude lokalni minimum je da  $\nabla f(x) \cong 0$ , jer dane globalne metode i perturbacija Hessianove matrice u pozitivno definitnu, osiguravaju da, ako postoji konvergencija, tada se konvergira prema minimumu. Međutim, sam uvjet  $\nabla f(x) \cong 0$ , koji testira

$$\|\nabla f(x_+)\| \leq \varepsilon \quad (6.2)$$

ovisi o skaliranju i  $f$  i  $x$  te nije adekvatan izbor. Primjerice, ako je  $\varepsilon = 10^{-3}$  i  $f$  je unutar  $[10^{-7}, 10^{-5}]$ , tada je vrlo vjerojatno da će bilo koja vrijednost od  $x$  zadovoljavati (6.2). S druge strane, ako je  $f \in [10^5, 10^7]$ , uvjet (6.2) je možda prestrogi. Isto tako, ako je  $x$  nekonzistentno skaliran, npr.  $x_1 \in [10^6, 10^7]$  i  $x_2 \in [10^{-1}, 1]$ , tada (6.2) varijable ne tretira

jednako. Čest odabir je tada primjena testa

$$|\nabla f(x_+)^T \nabla^2 f(x_+)^{-1} \nabla f(x_+)| \leq \varepsilon, \quad (6.3)$$

invarijantnog na bilo koju linearnu transformaciju neovisnih varijabli i skaliranja od  $x$ , ali i dalje ovisno o skaliranju funkcije  $f$ .

Dennis, Schnabel [1] daju direktniju modifikaciju uvjeta (6.2). Uvodi se relativan gradijent od  $f$  u točki  $x$  dan sa

$$\begin{aligned} \text{relgrad}(x)_i &= \frac{\text{relativna stopa promjene u } f}{\text{relativna stopa promjene od } x_i} \\ &= \lim_{\delta \rightarrow 0} \frac{\frac{f(x+\delta e_i) - f(x)}{f(x)}}{\frac{\delta}{x_i}} \\ &= \frac{\nabla f(x)_i x_i}{f(x)} \end{aligned} \quad (6.4)$$

i provodi se test

$$\|\text{relgrad}(x_+)\|_\infty \leq \text{gradtol}, \quad (6.5)$$

gdje  $\text{gradtol} > 0$  određuje toleranciju pogreške završetka algoritma. Tolerancija pogreške uglavnom predstavlja korisnikovu ideju rješenja problema, a ovdje se za dane algoritme preporuča  $\text{gradtol} = \text{macheps}^{1/3}$ .

Test (6.5) je neovisan o promjenama veličina  $i$  od  $f$  i od  $x$ , ali nedostatak je u tome što ideja o relativnoj promjeni u  $x_i$  ili  $f$  ne funkcionira ako su  $x_i$  ili  $f$  jako blizu 0. Nedostatak se rješava zamjenom  $x_i$  i  $f$  sa  $\max\{|x_i|, \text{typ}x_i\}$  i  $\max\{|f(x)|, \text{typ}f\}$ , respektivno u (6.4), gdje je  $\text{typ}f$  korisnikova procjena veličine  $f$ . Rezultirajući test, korišten i u danim algoritmima, je tada

$$\max_{1 \leq i \leq n} \left| \frac{\nabla f(x)_i \max\{|(x_+)_i|, \text{typ}x_i\}}{\max\{|f(x_+)|, \text{typ}f\}} \right| \leq \text{gradtol}. \quad (6.6)$$

Sljedeće pitanje koje se postavlja je: je li algoritam zastao ili je iskonvergirao? U tom slučaju, sljedeći gornju diskusiju, mjeri se relativna promjena u uzastupnim vrijednostima u  $x_i$  sa

$$\text{rel}x_i = \frac{|(x_+)_i - (x_c)_i|}{\max\{|(x_+)_i|, \text{typ}x_i\}}, \quad (6.7)$$

i dan je test

$$\max_{1 \leq i \leq n} \left| \frac{|(x_+)_i - (x_c)_i|}{\max\{|(x_+)_i|, \text{typ}x_i\}} \right| \leq \text{steptol}, \quad (6.8)$$

gdje je  $\text{steptol} > 0$  tolerancije pogreške za koju se smatra da je skalirana udaljenost između dvije uspješne iteracije dovoljno blizu 0, pri čemu bi  $\text{steptol}$  trebao biti najmanje barem

$10^{-d}$ , gdje je  $d$  broj željenih ispravnih decimala u rješenju  $x_*$ . Algoritam je zaustavljen kada je zadovoljen uvjet 6.8, a ukoliko je *septomol* prevelik, algoritam bi mogao završiti prerano, stoga se u našim algoritmima preporuča vrijednost  $septomol = macheps^{2/3}$ .

U iterativnim metodama važno je ograničiti broj iteracija, radi uštede vremena i troška procjene. U minimizacijskim problemima je poželjno provjeriti divergenciju iteracije  $x_k$  do koje može doći ukoliko je  $f$  neograničena odozdo ili asimptotski odozgo pristupa konačnoj donjoj granici. Za testiranje divergencije, korisnik unosi maksimalnu duljinu koraka  $maxstep > 0$ , koji osim što detektira divergenciju, sprječava napuštanje područja interesa. Poželjno je da bude dovoljno malen da to izbjegne i dovoljno velik da bude veći od svake očekivane veličine koraka. U danim algoritmima preporučena je vrijednost  $maxstep = 10^3 \max\{\|D_{x_0}x_0\|_2, \|D_x\|_2\}$ . Ukoliko su pet uzastopna koraka maksimalne duljine, algoritam staje.

U slučaju sustava nelinearnih jednadžbi najprije se testira rješava li  $x_+$  aproksimativno dani problem, odnosno vrijedi li  $F(x_+) \cong 0$ . Ponovno, zbog skaliranja, nije dovoljan test  $\|F(x_+)\| \leq \varepsilon$ . Međutim, obzirom da je  $(D_F)_{ii} = 1/typf_i$  izabrana da bude veličine 1 u točki ne previše blizu korijena, zadovoljavajuć je test

$$\|D_FF\|_\infty \leq fntol,$$

gdje je  $fntol > 0$  tolerancija pogreške za koju se skalirana funkcija  $D_FF(x)$  smatra da je dovoljno blizu 0. Algoritam se zaustavlja ako je maksimalna komponenta funkcije  $D_FF(x) \leq fntol$ . Ponovno se predlaže vrijednost  $fntol = macheps^{1/3}$ .

Test je li algoritam iskonvergirao ili je zastao u  $x_+$  je isti kao i kod minimizacijskog problema, (6.7) i (6.8), kao i test ograničenja broja iteracija i test divergencije, iako je manje vjerojatno da algoritam divergira kod rješavanja  $F(x) = 0$ . Moguće je da algoritam zapne u traženju lokalnog minimuma odgovarajuće minimizacijske funkcije  $f = \frac{1}{2}\|D_FF\|_2^2$ , gdje je  $F \neq 0$ . Iako test konvergencije, (6.7) i (6.8), u tom slučaju zaustavlja algoritam, poželjno je i eksplicitno provjeriti je li gradijent od  $f$  u  $x_+$  blizu 0, primjenom relativnih mjera gradijenta analogno (6.6). Ukoliko algoritam dosegne lokalni minimum od  $\|D_FF\|_2^2$  za koji je  $F \neq 0$ , potrebno ga je ponovo pokrenuti počevši u nekoj drugoj početnoj točki.

### 6.3 Testiranje

Nakon napisanog računalnog programa za nelinearne jednadžbe ili minimizacijske probleme, poželjno ih je testirati i vidjeti rade li pravilno i kakvi su u usporedbi s drugim softverima koji rješavaju isti problem.

Najprije je potrebno testirati radi li program ispravno, što za dane algoritme nije trivijalan zadatak. Predlaže se korištenje modularne procedure testiranja, pri čemu se najprije testira pojedini napisani modul, zatim dijelovi formirani tim modulima i na kraju cijeli

program. Prednost modularnog testiranja je lakše uočavanje mogućih grešaka programa, a nedostatak je što nekad nije jasno kako konstruirati početne unose podataka testnih modula, poput modula modificiranog trust region algoritma. Savjet je da se počne s podacima jednostavnijeg problema, poput dvodimenzionalnog problema sa identitetom ili dijagonalnom Jacobijevom ili Hessianovom matricom, a zatim pokušati s nešto kompleksnijim problemom.

Nakon što sve komponente rade ispravno, poželjno je program testirati na različitim nelinearnim problemima, što ima dvije svrhe: provjerava radi li cijeli program ispravno i promatra njegove preformanse na nekim standardnim problemima.

U nastavku slijedi par testnih primjera primjene izvedenih algoritama za problem bezuvjetne minimizacije. Za svaki će primjer funkcije biti prikazani dobiveni podaci o aproksimaciji minimuma, vrijednosti funkcije, broju iteracija, vremenu izvršavanja te uspješnosti pronalaska rješenja. U svakom primjeru su korištene vrijednosti  $\alpha = 10^{-4}$ ,  $gradtol = macheps^{1/3}$  i  $steptol = macheps^{2/3}$  te su provedeni testovi primjene backtracking line search algoritma (bez primjene 2. Wolfe-ovog uvjeta), modifikacije line search algoritma (line search s uključenim 2. Wolfovim uvjetom), kao i trust region algoritma s hook i s double dogleg korakom iteracije.

**Primjer 6.3.1.** Prvi testni primjer je funkcija zadana u primjeru 3.1.1. Neka je

$$f(x_1, x_2) = x_1^4 + x_1^4 + x_2^4$$

i početna točka iteracije  $x_0 = (1, 1)^T$ . Dobiveni su sljedeći rezultati:

Metoda	Line search	Modificirani line search	Hook trust region	Double dogleg trust region
Broj iteracija	5	5	5	5
Rješenje $x_*$	2.5968261E-10 6.8157555E-37	2.5968261E-10 6.8157555E-37	2.5968261E-10 6.8157555E-37	2.5968261E-10 6.8157555E-37
$f(x_*)$	6.7435000E-20	6.7435000E-20	6.7435000E-20	6.7435000E-20
Procijenjeno vrijeme izračuna u sekundama	0.018410	0.012955	0.016337	0.015642

Slika 6.1: Tablica rezultata primjera 6.3.1

Egzaktni minimum dane funkcije je u točki  $x_* = (0, 0)^T$ . Sve metode su vrlo brzo iskonvergirale dajući gotovo jednak rezultat.

**Primjer 6.3.2.** Neka je dana Wood funkcija

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$$

s početnom točkom  $x_0 = (3, -1, -3, -1)^T$ .

Metoda	Line search	Modificirani line search	Hook trust region	Double dogleg trust region
Broj iteracija	31	31	31	60
Rješenje $x_*$	1.0002644 1.0004960 0.9998400 0.9996617	1.0002644 1.0004960 0.9998400 0.9996617	0.9999997 0.9999993 1.0000003 1.0000006	1.0000015 1.0000030 0.9999984 0.9999968
$f(x_*)$	5.5186000E-07	5.5185530E-07	4.3270000E-13	8.9106716E-12
Procijenjeno vrijeme izračuna u sekundama	0.057762	0.065696	0.114398	0.110204

Slika 6.2: Tablica rezultata primjera 6.3.2

Ponovno su sve metode iskonvergirale prema rješenju  $x_* = (1, 1, 1, 1)^T$ , pri čemu je dogleg metoda primijenila najviše iteracija, što kod trust region metode, između ostalog, ovisi o odabiru regije povjerenja, odnosno parametru  $\delta$ .

**Primjer 6.3.3.** Neka je  $x_0 = (-1.2, 1, -1.2)^T$  početna točka za danu funkciju

$$f(x) = 100(x_2 - x_1)^2 + (1 - x_1)^2 + 100(x_3 - x_2^2)^2 + (1 - x_2)^2.$$

Metoda	Line search	Modificirani line search	Hook trust region	Double dogleg trust region
Broj iteracija	2	2	239	14
Rješenje $x_*$	1.0000004 1.0000004 1.0000008	1.0000004 1.0000004 1.0000008	0.9999957 0.9999957 0.9999914	1.0000000 1.0000000 1.0000000
$f(x_*)$	3.5416359E-13	3.5416359E-13	3.6856522E-11	2.3766427E-18
Procijenjeno vrijeme izračuna u sekundama	0.011741	0.009412	0.562487	0.027185

Slika 6.3: Tablica 1 rezultata primjera 6.3.3

Kao i u gornjem primjeru, sve metode su iskonvergirale prema rješenju  $x_* = (1, 1, 1)^T$ , od kojih je najbrže iskonvergirala modificirana metoda line searcha. Međutim, kao što je ranije spomenuto, brzina i broj iteracija trust region metode ovisi i o odabiru parametra  $\delta$ , što se u ovom primjeru može vidjeti iz sljedeće tablice

Metoda	$\delta=0.1$		$\delta=2$	
	Hook trust region	Double dogleg trust region	Hook trust region	Double dogleg trust region
Broj iteracija	237	12	2	2
Rješenje $x_*$	0.9999957	1.0000000	1.0000004	1.0000004
	0.9999957	1.0000000	1.0000004	1.0000004
	0.9999914	1.0000000	1.0000008	1.0000008
$f(x_*)$	3.6747899E-11	1.2137161E-16	3.5416359E-13	3.5416359E-13
Procijenjeno vrijeme izračuna u sekundama	0.580274	0.026066	0.011599	0.011599

Slika 6.4: Tablica 2 rezultata primjera 6.3.3

**Primjer 6.3.4.** Posljednji primjer, s početnom točkom  $x_0 = (0.1, 0.1)^T$ , je funkcija

$$f(x) = 20 + x_1^3 + x_2^3 - 10(\cos(2\pi x_1) + \cos(2\pi x_2)),$$

za koju su sve metode relativno brzo iskonvergirale prema rješenju  $x_* = (0, 0)^T$ .

Metoda	Line search	Modificirani line search	Hook trust region	Double dogleg trust region
Broj iteracija	12	12	4	5
Rješenje $x_*$	-1.8512903E-09	-1.8512903E-09	-2.2410245E-11	-6.2360750E-13
	-1.8512903E-09	-1.8512903E-09	-2.2410245E-11	-6.2360750E-13
$f(x_*)$	3.5527137E-15	3.5527137E-15	0.0000000E+00	0.0000000E+00
Procijenjeno vrijeme izračuna u sekundama	0.014114	0.017775	0.026802	0.017694

Slika 6.5: Tablica rezultata primjera 6.3.4

U danim primjerima, sve metode postižu približno jednake rezultate, koji se uglavnom razlikuju ovisno o postavljanju inicijalnih uvjeta i parametara.



# Bibliografija

- [1] J.E.Dennis, Jr., R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1996.
- [2] P. E. Gill, W. Murray, M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.
- [3] A. A. Goldstein, *Constructive Real Analysis*, Harper and Row, New York, 1967.
- [4] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Review 11 (1969), 226–235.
- [5] P. Wolfe, *Convergence conditions for ascent methods. II: Some corrections*, SIAM Review 13 (1971), 185-188.
- [6] L. Armijo, *Minimization of functions having Lipschitz-continuous first partial derivatives*, Pacific J. Math. 16 (1966), 1-3.
- [7] J.E. Dennis, Jr., J. J. More, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp. 28 (1974), 549-560.
- [8] D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. Stat. Comp. 2 (1981), 186-197.
- [9] K. Levenberg, *A method for the solution of certain problems in least squares* Quart. Appl. Math. 2 (1944), 164-168.
- [10] D. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, SIAM J. Appl. Math. II (1963), 431-441.
- [11] S. M. Goldfeldt, R. E. Quandt., H. K. Trotter, *Maximization by quadratic hill-climbing*, Econometrica 34 (1966), 541-551.

- [12] M. D. Hebden, *An algorithm for minimization using exact second derivatives*, Theoretical Physics Division A.E.R.E. Harwell, Engleska, 1973.
- [13] J. J. Móre, *The Levenberg-Marquardt algorithm: implementation and theory*, Lecture Notes in Mathematics 630 - Numerical Analysis (G. A. Watson), Springer Verlag, Berlin, 1977., 105-116.
- [14] C. H. Reinsch, *Smoothing by spline functions. II*, Numer. Math. 16 (1971), 451-454.
- [15] M.J.D. Powell, *A hybrid method for nonlinear equations*, Numerical Methods for Nonlinear Algebraic Equations (P. Rabinowitz), Gordon and Breach Science, London, 1970., 87-114.
- [16] J. E. Dennis, Jr., H. H. W. Mei, *Two new unconstrained optimization algorithms which use function and gradient values* J. Optim. Theory .Appl. 28 (1979), 453-482.
- [17] M. Bierlaire, Descent methods and line search: preconditioned steepest descent, [https://www.youtube.com/watch?v=3wh\\_TLzuiRI](https://www.youtube.com/watch?v=3wh_TLzuiRI) (travanj 2021.)
- [18] M. Bierlaire, Descent methods and line search: first Wolfe condition, <https://www.youtube.com/watch?v=X4Pjd-1R-jI&t=15s> (travanj 2021.)
- [19] M. Bierlaire, Descent methods and line search: second Wolfe condition, <https://www.youtube.com/watch?v=5upFcYJqSwo> (travanj 2021.)
- [20] M. Bierlaire, Descent methods with line search: Newton method with line search, <https://www.youtube.com/watch?v=x1wMciVA6Xc> (travanj 2021.)

# Sažetak

U ovom radu predstavljena je osnovna Newtonova metoda za rješavanje problema bezuvjetne minimizacije i nelinearnih jednadžbi, njena konvergencija, nedostatci i neka bitna svojstva za daljni razvoj metode.

Glavni problem metode je nedostatak globalne konvergencije što je razlog uvođenja globalnih metoda poput line search i trust region algoritama. Osnovna razlika line search i trust region pristupa je u tome što line search traži smjer napretka iteracije (smjer silaska) i odabire duljinu koraka u tom smjeru, dok trust region odabire područje regije povjerenja (područje maksimalne duljine koraka) i odabire točku napretka unutar te regije.

Nadalje, iskazane su prednosti i nedostatci globalnih metoda, od kojih je glavni nedostatak brza konvergencija, zbog čega se uvodi kvazi-Newtonov algoritam. Glavni zadatak je inkorporiranje lokalno brze Newtonove metodu s globalno konvergentnim metodama kako bi se iskoristile prednosti, a izbjegli nedostatci, pojedine metode. Svi algoritmi se prvo formiraju za rješavanje problema bezuvjetne minimizacije, a zatim se preslikavaju na probleme nelinearnih jednadžbi. Naposljetku su definirani problemi skaliranja i potrebni kriteriji zaustavljanja kako bi se algoritmi mogli provesti u praksi.

Na kraju rada je provedeno testiranje nekoliko jednostavnih problema bezuvjetne minimizacije, ali je pritom teško odgovoriti na pitanje koji je algoritam bolji jer, primjena određene metode, ovisi i o zadanim početnim uvjetima i parametrima, kao i o željenom rezultatu. Međutim, danas je razvijeno mnogo modificiranih oblika postavljenih algoritama i najčešće se kod rješavanja problema primjenjuju njihove kombinacije.

# Summary

In this thesis, we have introduced Newton's method for unconstrained minimization and nonlinear equations, its convergence, shortcomings and basic properties for further development of methods.

Main problem of Newton's method is lack of global convergence and that is why we introduce global methods like line search and trust region algorithms. Difference between line search and trust region approach is that line search finds direction of improvement and then selects the step length in descent direction, and trust region first selects the trusted region of maximum allowed step and then finds a point of improvement in the region.

Furthermore, the advantages and disadvantages of global methods are shown. Main defect is lack of fast convergence, which is a reason for development of quasi-Newton's method. Main purpose is to incorporate locally fast convergent Newton's method with globally convergent methods in order to take their advantages, and avoid their disadvantages. All algorithms are formed to solve problems of unconstrained optimization, and then discussed their modifications for nonlinear equations. Lastly, scaling and stopping criteria, important for real practice, are discussed.

At the end, several examples of simple optimization problems are tested. It is not easy to say which method is better to apply, because its performance and execution depends on several reasons, like starting point or parameter setting. However, there are many different modified algorithms today, and its combinations are being used for solving the problems.

# Životopis

Rođena 25.siječnja 1994. godine u Čakovcu, gdje sam po završetku osnovne škole započela srednješkolsko obrazovanje u Gimnaziji Čakovec, danas zvanj Gimnazija Josipa Slavenkog Čakovec. Nakon položene državne mature 2012., upisujem Preddiplomski sveučilišni studij matematike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta u Zagrebu i na istom fakultetu, stjecanjem titule prvostupnika 2017. godine, nastavljam s Diplomskim sveučilišnim studijem Financijske i poslovne matematike.