Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku

Milan Milinčević

Quasi-Newtonove metode

Završni rad

Osijek, 2016.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku

Milan Milinčević

Quasi-Newtonove metode

Završni rad

Voditelj: izv. prof. dr. sc. Kristian Sabo

Osijek, 2016.

Sadržaj

1. Quasi-Newtonove metode

1.1. Sažetak

U ovom radu ćemo se baviti rješavanjem jednadžbe $f(x) = 0$ pomoću Newtonove metode tangente te metode sekante, kada nam je funkcija $f : \mathbb{R} \to \mathbb{R}$. Za višedimenzionalni problem problem koristit ćemo Newtonovu metodu. Zbog numeričke nestabilnosti Newtonove metode, za višedimenzionalni problem ćemo također koristiti jednu od najpoznatijih tzv. Quasi-Newtonovih metoda, Broydenovu metodu. Objema metodama ćemo rješiti isti primjer te ćemo vidjeti prednosti i nedostatke svake od njih. Reći ćemo nešto još i o dvjema najpoznatijim optimizacijskim Quasi-Newton metodama; Broyden-Fletcher-Goldfarb-Shanno te Davidon-Fletcher-Powell metodi. Pomoću njih ćemo tražiti minimum funkcije $f : \mathbb{R}^n \to \mathbb{R}$.

1.2. Ključne riječi

Newtonova metoda tangente, metoda sekante, Newtonova metoda, Jacobijan, Broydenova metoda, Hessijan, Broyden-Fletcher-Goldfarb-Shanno metoda, Davidon-Fletcher-Powell metoda.

2. Quasi-Newton methods

2.1. Abstract

In this work we will be interested in solving equation $f(x) = 0$ using Newton's tangent method and secant method, when we have function $f : \mathbb{R} \to \mathbb{R}$. For multidimensional problem we will use Newton's method. Because of numerical instability of Newton's method, for multidimensional problem we shall also use one of the best known so-called Quasi-Newton's method, the Broyden's method. We will solve the same example with both methods and compare adventages and disadvantages each of them. We shall say something about two best known optimization Quazi-Newton's methods; Broyden-Fletcher-Goldfarb-Shanno and Davidon-Fletcher-Powell method. Using them, we will try to minimize function $f : \mathbb{R}^n \to \mathbb{R}$.

2.2. Key words

Newton's tangent method, secant method, Newton's method, Jacobian, Broyden's method, Hessian, Broyden-Fletcher-Goldfarb-Shanno method, Davidon-Fletcher-Powell method.

Poglavlje 1

Uvod

U ovom radu, tražit ćemo rješenje jednadžbe:

$$f(x) = 0$$

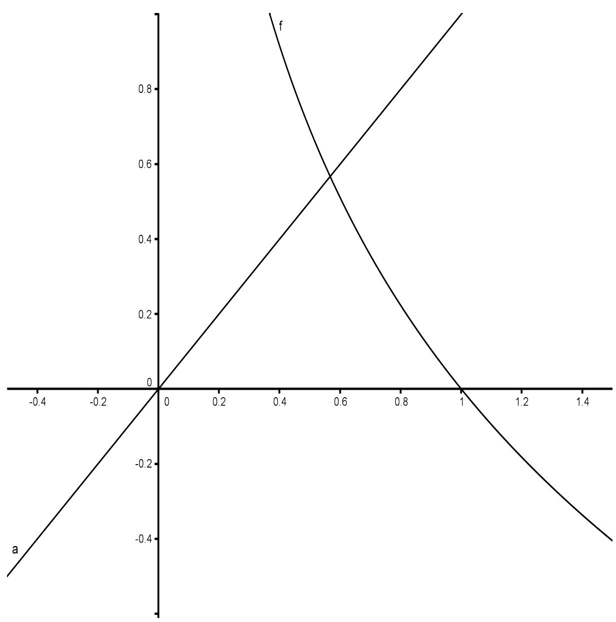
pri čemu je funkcija $f : \mathbb{R}^n \to \mathbb{R}^n$. Najprije analiziramo problem rješavanja ove jednadžbe u slučaju $n = 1$, odnosno pretpostavljamo da je $f : I \to \mathbb{R}$, gdje je $I = [a, b] \subseteq \mathbb{R}$. Najznačajnije dvije metode za rješavanje ovog problema su Newtonova metoda i metoda sekante. O njima ćemo nešto više reći jer bez njih ne bismo imali niti iterativne metode za višedimenzionalni problem.
Najpoznatija iterativna metoda za rješavanje naše jednadžbe je Newtonova metoda. Međutim, u svakoj iteraciji Newtonove metode trebamo iznova računati matricu parcijalnih derivacija (Jacobijan), a to nije numerički stabilno. Zbog toga postoje metode koje su pouzdanije. Jedna od njih je Broydenova metoda, koja je jedna od najpoznatijih tzv. Quasi-Newtonovih. U njoj aproksimiramo Jacobijan, no i to ima svoju cijenu, a to je brzina konvergencije. Budući da je rješenje ovakvog iterativnog postupka dovoljno blizu pravom rješenju, česta je upotreba upravo Broydenove metode unatoč tome što je potrebno nešto više iteracija pri dolasku do one točke do koje Newtovom metodom dođemo u manje koraka.
Dotaći ćemo se još i dviju optimizacijskih Quasi-Newtonovih metoda, Broyden-Fletcher-Goldfarb-Shanno te Davidon-Fletcher-Powell metode. Te dvije metode koristimo za traženje minimuma funkcije. Metode za minimizaciju su nešto kompliciranije od običnih Quasi-Newton metoda za rješavanje jednadžbe $f(x) = 0$ jer moraju zadovoljavati neka svojstva koja za iterativne metode za rješavanje navedene jednadžbe nisu nužna, no o tome ćemo nešto kasnije.
Svakom od tih metoda riješit ćemo jedan primjer kako bismo vidjeli način na koji one funkcioniraju, te da u praksi usporedimo prednosti i nedostatke svake od njih. Primjere ćemo riješiti u MATLAB-u i navesti pripadne programske kodove.

Poglavlje 2

Motivacija

Za zadanu funkciju $f : \mathbb{R}^n \to \mathbb{R}^n$ zanima nas rješenje jednadžbe $f(x) = 0$. Često to nije tako jednostavno. Promotrimo to na sljedećem primjeru.

Primjer 1 *Neka je funkcija $f : \mathbb{R} \to \mathbb{R}$ zadana s $f(x) = x + lnx$. Želimo riješiti jednadžbu $f(x) = 0$, odnosno $x + lnx = 0$. Analogan zapis je sljedeći: $x = -lnx$. Rješenje te jednadžbe je točka u kojoj se sijeku grafovi $y_1 = x$ i $y_2 = -lnx$. Nacrtajmo oba grafa.*



Budući da ovakve jednadžbe ne možemo riješiti egzaktno, do rješenja pokušavamo doći primjenom numeričkih iterativnih postupaka. U ovom radu, za rješavanje jednadžbe $f(x) = 0$, gdje je $f : \mathbb{R} \to \mathbb{R}$ koristit ćemo Newtonovu metodu tangenti, te metodu sekanti, dok ćemo općenito za funkcije $f : \mathbb{R}^n \to \mathbb{R}^n$ koristiti Newtonovu te Broydenovu metodu.

Poglavlje 3

Newtonova metoda tangente i metoda sekante

1. Newtonova metoda tangente

Na početku ćemo nešto reći o Newtonovoj metodi tangente. Ova metoda služi nam za rješavanje jednadžbe $f(x) = 0$, pri čemu je funkcija f realna neprekidno diferencijabilna funkcija definirana na segmentu $I = [a, b]$. Ako je f neprekidna funkcija na I, te ako vrijedi $f(a) \cdot f(b) < 0$, onda postoji barem jedna točka $c \in I$ td. je $f(c) = 0$ (vidi [4]). Ako je još i f' stalnog predznaka na segmentu I, onda je c jedina nultočka funkcije f na I.

Pretpostavimo da je $f'(x) \neq 0, \forall x \in I$. Izaberimo neku točku $x_0 \in I$ i povucimo tangentu na graf funkcije f kroz točku $(x_0, f(x_0))$. Jednadžba tangente kroz zadanu točku je:

$$f(x) - f(x_0) = f'(x_0)(x - x_0).$$

Kako nam ova metoda služi za rješavanje jednadžbe $f(x) = 0$ slijedi:

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Dobili smo točku x u kojoj tangenta siječe $x - os$. Označimo tu točku s x_1 i povucimo sada tangentu na graf funkcije kroz točku $(x_1, f(x_1))$. Dobivenu točku označimo s x_2.

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

Nastavljajući ovaj postupak povlačeći tangente kroz točke $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ dobijemo iterativni postupak koji je zadan sljedećom formulom ([4]):

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, \ldots \quad .$$

Teorem 1 **([4])** *Neka je $f : I \to \mathbb{R}, I = [a, b]$ td. je $f(a) \cdot f(b) < 0$, te neka je f'' neprekidna na I. Neka su $f' i f''$ stalnog predznaka na I. Ako je $x_0 \in I$ td. $f(x_0) \cdot f''(x_0) > 0$, onda niz definiran s:*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, \ldots$$

konvergira prema jedinstvenom rješenju ξ jednadžbe $f(x) = 0$ i vrijedi:

$$|\xi - x_n| \leq \frac{M_2}{2m_1}(x_n - x_{n-1})^2$$

$$|\xi - x_{n+1}| \leq \frac{M_2}{2m_1}(\xi - x_n)^2$$

pri čemu je: $m_1 := \min_{x \in I} |f'(x)|$, $M_2 := \max_{x \in I} |f''(x)|$.

2. Metoda sekante

Ova metoda, kao i Newtonova metoda tangenti služi za rješavanje jednadžbe $f(x) = 0$, pri čemu je funkcija f realna neprekidna funkcija definirana na segmentu $I = [a, b]$. Velik je uvjet tražiti derivabilnu funkciju jer metodu želimo primjenjivati na što većem skupu funkcija. Zbog toga ćemo uzeti 2 početne aproksimacije $x_0, x_1 \in I$ te kroz točke $(x_0, f(x_0)), (x_1, f(x_1))$ povučemo sekantu na graf funkcije f (vidi [4]).

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) = 0$$

Sredimo ovaj izraz, s time da ćemo točku u kojoj sekanta siječe x-os označiti s x_2:

$$x_2 = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}.$$

Sada uzmemo aproksimacije x_1 i x_2 te ponovimo isti postupak. Iterativnim ponavljanjem dobijemo sljedeći algoritam:

$$x_{n+1} = \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}, \quad n = 1, 2, \ldots \quad .$$

Na taj smo način izostavili uvjet o derivabilnosti, no za to smo platili određenu cijenu; gubitak na brzini konvergencije. Brzina konvergencije Newtonove metode tangenti je kvadratna, a metode sekante ipak nešto manja i iznosi $\frac{1+\sqrt{5}}{2} \approx 1.618$ (vidi [4]).

Poglavlje 4

Newtonova i Quasi-Newtonove metode za rješavanje sustava nelinearnih jednadžbi

1. Newtonova metoda

Neka je $F = (f_1, f_2, \ldots, f_n) : \mathbb{R}^n \to \mathbb{R}^n$ neprekidno diferencijabilna funkcija, želimo naći $x \in \mathbb{R}^n$ td. je $F(x) = 0$.

Definicija 1 *Neka je $f = (f_1, f_2, \ldots, f_n) : \mathbb{R}^n \to \mathbb{R}^n$ neprekidno diferencijabilna funkcija i $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ tada matricu*

$$J = \frac{df}{dx} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

nazivamo Jacobijan ili Jacobijeva matrica.

Izaberimo početnu aproksimaciju $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ i svaku od funkcija $f_1, \ldots, f_n$ razvijmo u Taylorov red oko točke $x^{(0)}$ do uključivo prvog člana ([4]):

$$\tilde{f}_i(x) = f_i(x^{(0)}) + \sum_{j=1}^{n} \frac{\partial f_i(x^{(0)})}{\partial x_j}(x_j - x_j^{(0)}), \quad i = 1, \ldots, n.$$

Sada ćemo umjesto jednadžbe $F(x) = 0$ rješavati sustav $\tilde{f}_i(x) = 0$, što u matričnom zapisu izgleda:

$$J(x^{(0)})s(x^{(0)}) = -F(x^{(0)})$$

pri čemu je $J(x^{(0)})$ Jacobijan u točki $x^{(0)}$, $s(x^{(0)}) = (x_1 - x_1^{(0)}, \ldots, x_n - x_n^{(0)})$, a $F(x^{(0)})$ vrijednost funkcije F u točki $x^{(0)}$. Tada nam je nova aproksimacija rješenja:

$$x^{(1)} = x^{(0)} + s(x^{(0)}).$$

Općenito, dobijemo iterativni postupak:

$$x^{(k+1)} = x^{(k)} + s(x^{(k)}), \quad k = 0, 1, \ldots$$

pri čemu $s(x^{(k)})$ dobijemo kao rješenje sustava:

$$J(x^{(k)})s(x^{(k)}) = -F(x^{(k)}).$$

Pretpostavimo li da je Jacobijan regularan, dobit ćemo:

$$s(x^{(k)}) = -\big(J(x^{(k)})\big)^{-1}F(x^{(k)}), \quad k = 0, 1, \ldots$$

odnosno, kad uvrstimo:

$$x^{(k+1)} = x^{(k)} - \big(J(x^{(k)})\big)^{-1}F(x^{(k)}), \quad k = 0, 1, \ldots$$

što nas podsjeća na Newtonovu metodu tangente. Međutim, izračunavanje inverzne matrice Jacobijana nas puno košta, pa se ovaj iterativni postupak rjeđe koristi(vidi [4]).

Ako je u svakoj iteraciji Jacobijan nesingularan, te ako početnu aproksimaciju izaberemo dovoljno blizu rješenja, Newtonova metoda kvadratnom brzinom konvergira prema rješenju sustava. Drugim riječima, nismo osigurani da će ova metoda uvijek konvergirati (vidi [2]).

Primjer 1 *Riješimo Newtonovom metodom sljedeći sustav jednadžbi:*

$$2x^2 + y^2 - 1 = 0$$

$$x^3 + 6x^2y - 1 = 0$$

Tada je:

$$f(x,y) = \begin{bmatrix} 2x^2 + y^2 - 1 \\ x^3 + 6x^2y - 1 \end{bmatrix} \qquad J(x,y) = \begin{bmatrix} 4x & 2y \\ 3x^2 + 12y & 6x^2 \end{bmatrix}.$$

Izaberimo početnu aproksimaciju: $x^{(0)} = (0.5, 0.5)$. *Rješavat ćemo iterativno primjer dok god nam je* $||f(x^{(k)})||_\infty > eps$, *pri čemu je* $eps > 0$, *u našem slučaju strojna preciznost računala. Uključit ćemo naredbu* ***tic*** *u MATLAB-u da vidimo koliko nam vremena treba da dođemo do rješenja, te ćemo ga usporediti s vremenom izvršavanja istog primjera Broydenovom metodom.*

Pripadni MATLAB kod za rješenje primjera izgleda:

```
tic
format long
x0=0.5;
y0=0.5;
syms x y;

f1(x,y)=2*x^2+y^2-1;
f2(x,y)=x^3+6*x^2*y-1;
J1(x,y)=jacobian([2*x^2+y^2-1], [x, y]);
J2(x,y)=jacobian([x^3+6*x^2*y-1], [x, y]);

while max(abs(f1(x0,y0)),abs(f2(x0,y0)))>eps
    a=J1(x0,y0);
    b=J2(x0,y0);
    q=double(a(1)/(b(1)*a(2)+b(2)*a(1))*(f2(x0,y0)-f1(x0,y0)*b(1)/a(1)));
    p=double(f1(x0,y0)/a(1)-a(2)/a(1)*q);
    x=[x0 y0]
    x1=x0-p;
    y1=y0-q;
    x0=x1;
    y0=y1;
end
toc
```

Pokrenemo kod i dolazimo do rješenja: $x^{(11)} = (0.68659, 0.23912)$, *pri čemu vrijeme potrebno za izvršavanje programa iznosi* 2.49 *sekundi.*

2. Broydenova metoda

Kod Newtonove metode jako je skupo računati Jacobijan u svakoj iteraciji, pa zbog toga uvodimo tzv. Quasi-Newtonove metode, kako bismo aproksimirali Jacobijan. Jedna od najpoznatijih takvih metoda je Broydenova, koja je nastala generalizacijom metode sekante.

Neka nam vrijede iste oznake kao kod Newtonove metode. Kao što smo već pokazali, aproksimaciju funkcije F u okolini točke $x^{(0)}$ možemo zapisati kao:

$$\tilde{F}(x) = F(x^{(0)}) + J(x^{(0)})(x - x^{(0)})$$

Označimo s $\tilde{B}$ aproksimaciju Jacobijana u točki $x^{(0)}$. Tada vrijedi:

$$\tilde{F}(x) = F(x^{(0)}) + \tilde{B}(x^{(0)})(x - x^{(0)}).$$

Uz oznake $s := x - x^{(0)}$ i $y := F(x) - F(x^{(0)})$ vrijedi:

$$\tilde{B}s = y.$$

Neka je B poznata aproskimacija od J. Broydenova pretpostavka je da se $\tilde{B}$ razlikuje od B samo u smjeru od s, tj. ako je $< z, s >= 0$ onda je $\tilde{B}z = Bz$ (vidi [1]).

Provjerimo zadovoljava li $\tilde{B}$ definiran na sljedeći način: $\tilde{B} = B + \frac{(y-Bs)s^T}{<s,s>}$ prethodna 2 svojstva.

ispitajmo prvo vrijedi li $\tilde{B}s = y$.

$$\left(B + \frac{(y - Bs)s^T}{< s, s >}\right)s = y$$

$$Bs + \frac{(y - Bs)s^T s}{s^T s} = y$$

$$Bs + y - Bs = y$$

$$y = y$$

Ostaje nam još za pokazati da je $\tilde{B}z = Bz$ ako je $< z, s >= 0$.

$$\left(B + \frac{(y - Bs)s^T}{< s, s >}\right)z = Bz$$

$$Bz + \frac{(y - Bs)s^T z}{< s, s >} = Bz$$

$$Bz + \frac{(y - Bs)}{< s, s >} < z, s >= Bz$$

Kako je po pretpostavci $< z, s >= 0$ slijedi:

$$Bz = Bz.$$

Time smo dokazali i ovu tvrdnju, te smo dobili direktnu Broydenovu metodu:

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

pri čemu matrice $B_k \in \mathbb{R}^{n \times n}$ dobivamo rekurzivno iz formule:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T}{< s_k, s_k >}, \quad k = 0, 1, \dots$$

gdje su $y_k = F(x_{k+1}) - F(x_k)$ i $s_k = x_{k+1} - x_k$ (vidi [2]).
Računanje inverzne matrice nas košta i to ćemo izbjeći tako što ćemo umjesto $-B_k^{-1} F(x_k)$ rješavati:

$$B_k s_k = -F(x_k), \quad k = 0, 1, \dots$$

pa dobijemo:

$$x_{k+1} = x_k + s_k, \quad k = 0, 1, \dots \quad .$$

Primjer 1 *Riješimo Broydenovom metodom sljedeći sustav jednadžbi:*

$$2x^2 + y^2 - 1 = 0$$

$$x^3 + 6x^2 y - 1 = 0$$

Ovaj primjer smo već riješili Newtonovom metodom. Opet imamo istu funkciju:

$$f(x, y) = \begin{bmatrix} 2x^2 + y^2 - 1 \\ x^3 + 6x^2 y - 1 \end{bmatrix}.$$

No u ovom slučaju ne moramo izračunavati Jacobijan funkcije f u svakoj iteraciji nego ćemo ga aproksimirati matricama $B_k, \quad k = 0, 1, \dots$. Za nultu iteraciju, tj. matricu B_0 uzimamo jediničnu matricu I reda n. Uzmimo opet istu početnu aproksimaciju $x^{(0)} = (0.5, 0.5)$. Također ćemo uključiti naredbu ***tic*** *u MATLAB-u, da bismo izmjerili vrijeme potrebno za izvršavanje programa.*

Riješimo sada u MATLAB-u primjer:

```
tic
format long
x0=0.5;
y0=0.5;
syms x y;

f1(x,y)=2*x^2+y^2-1;
f2(x,y)=x^3+6*x^2*y-1;
B=eye(2);
kk=0;
x=zeros(17,2);

while max(abs(f1(x0,y0)),abs(f2(x0,y0)))>eps
    q=double(B(1,1)/(B(1,2)*B(2,1)+B(2,2)*B(1,1))*...
    (-f2(x0,y0)+f1(x0,y0)*B(2,1)/B(1,1)));
    p=double(-f1(x0,y0)/B(1,1)-B(1,2)/B(1,1)*q);
    x(kk+1,:)=[x0 y0];
    x1=x0+p;
    y1=y0+q;
    t0=f1(x1,y1)-f1(x0,y0);
    t1=f2(x1,y1)-f2(x0,y0);
    s=p^2+q^2;
    B=B+1/s*[t0-B(1,1)*p-B(1,2)*q; t1-B(2,1)*p-B(2,2)*q]*[p q];
    x0=x1;
    y0=y1;
    kk=kk+1;
end
toc
```

te dolazimo do rješenja $x^{(17)} = (0.68659, 0.23912)$. *Trebalo nam je čak* 6 *iteracija više da dođemo do istog rješenja kao kod Newtonove metode. Vrijeme potrebno za izvršavanje ovog programa je* 5.22 *sekunde. To je i očekivano jer smo već spomenuli kako je brzina konvergencije Newtonove metode veća nego u Broydenove.*

3. Broyden-Fletcher-Goldfarb-Shanno (BFGS) metoda

Dok Broydenova metoda služi za rješavanje sustava nelinearnih jednadžbi, ova Quasi-Newtonova metoda se koristi za optimizaciju funkcije. Broydenova metoda ne čuva neka strukturna svojstva potrebna za metode linijskim pretraživanjem [1] u optimizaciji (simetričnost i pozitvna definitnost) i mogla bi zapravo potaknuti konvergenciju prema lokalnom maksimumu. Iz tog su razloga Quasi-Newtonove optimizacijske metode kompliciranije od onih koje se koriste za rješavanje nelinearnih jednadžbi.

Definicija 1 *Neka je* $f = (f_1, f_2, \dots, f_n) : \mathbb{R}^n \to \mathbb{R}^n$ *funkcija koja ima neprekidnu drugu derivaciju na* $\mathbb{R}^n$ *i neka je* $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$*. Tada matricu:*

$$H = \begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & \cdots & \frac{\partial f_1}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f_n}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f_n}{\partial x_n^2} \end{bmatrix}$$

nazivamo Hesseova matrica ili Hessian.

BFGS je Quasi-Newtonova metoda koja određuje aproksimaciju od matrice H kao iterativni postupak. Pri čemu aproksimaciju matrice H dobivamo na sljedeći način:

$$H_{k+1} = H_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{(H_k s_k)(H_k s_k)^T}{s_k^T H_k s_k}, \quad k = 1, 2, \dots$$

gdje su $s_k := x_{k+1} - x_k$ i $y_k := \nabla(x_{k+1}) - \nabla(x_k)$ (vidi [3]).
Ova metoda je tzv. metoda sekante iz razloga što zadovoljava jednadžbu sekante:

$$H_{k+1} s_k = y_k, \quad k = 1, 2, \dots \quad .$$

Za n=1, tj. za funkcije $f : \mathbb{R} \to \mathbb{R}$ sve metode sekante reduciraju se to klasične metode sekante za nelinearnu jednadžbu $f'(x) = 0$, tj. vrijedi:

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

[1] Linijsko pretraživanje-jedno od 2 osnovna iterativna pristupa u pronalsku lokalnog minimuma funkcije $f : \mathbb{R}^n \to \mathbb{R}^n$

Primjer 1 *Odredimo minimum funkcije $f : \mathbb{R}^2 \to \mathbb{R}$ definirane s:*

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

Za početnu aproksimaciju uzet ćemo točku $x^{(0)} = (0.8, 2.7)$. Pripadni MATLAB kod izgleda:

```
tic
format long
x0=0.8;
y0=2.7;
syms x y;

f1(x,y)=(x+2*y-7)^2+(2*x+y-5)^2;
H=eye(2);
J1(x,y)=jacobian([(x+2*y-7)^2+(2*x+y-5)^2], [x, y]);

while max(abs(f1(x0,y0)))>eps
    j=J1(x0,y0);
    q=double(H(1,1)/(H(1,2)*H(2,1)+H(2,2)*H(1,1))*...
        (-j(2)+j(1)*H(2,1)/H(1,1)));
    p=double(-j(1)/H(1,1)-H(1,2)/H(1,1)*q);
    x=[x0 y0];
    x1=x0+p;
    y1=y0+q;
    t0=J1(x1,y1)-J1(x0,y0);
    ys=t0(1)*p+t0(2)*q;
    z=p*(H(1,1)*p+H(1,2)*q)+q*(H(2,1)*p+H(2,2)*q);
    H=H+1/ys*[p^2 p*q;p*q q^2]-1/z*[(H(1,1)*p+H(1,2)*q)^2 (H(1,1)*...
        p+H(1,2)*q)*(H(2,1)*p+H(2,2)*q);(H(1,1)*p+H(1,2)*q)*(H(2,1)*...
        p+H(2,2)*q) (H(2,1)*p+H(2,2)*q)^2];
    x0=x1;
    y0=y1;
end
toc
```

Točka minimuma iznosi: $x^{(9)} = (1, 3)$. Vrijeme potrebno da se program izvrši iznosi 8.13 *sekundi.*

4. Davidon-Fletcher-Powell (DFP) metoda

Problem optimizacije diferencijabilne funkcije $f : \mathbb{R}^n \to \mathbb{R}$ možemo gledati i kao problem rješavanja jednadžbe:

$$F(x) = 0$$

pri čemu je funkcija $F : \mathbb{R}^n \to \mathbb{R}^n$ td. je $F(x) = f'(x)$. Taj problem bismo mogli riješiti Newtonovom ili nekom drugom iterativnom metodom za rješavanje sustava nelinearnih jednadžbi, no postoje iterativne metode za rješavanje optimizacijskog problema; ranije spomenuta Broyden-Fletcher-Goldfarb-Shanno te Davidon-Fletcher-Powell metoda o kojoj ćemo sada reći nešto više.
Kao što smo već naveli, Davidon-Fletcher-Powell metodu koristimo za optimizaciju funkcije. Ova metoda je poznata kao prva Quasi-Newtonova metoda koja zadovoljava jednadžbu sekante za multidimenzionalni problem.
DFP metodom dobivamo aproksimacuju matrice Hessijana H kroz iterativni postupak, gdje aproksimacije matrice H dobivamo kroz sljedeći postupak:

$$H_{k+1} = H_k + \frac{(y_k - H_k s_k)y_k^T + y_k(y_k - H_k s_k)^T}{y_k^T s_k} - \frac{\big((y_k - H_k s_k)^T y_k\big) y_k y_k^T}{(y_k^T s_k)^2}, \quad k = 1, 2, \ldots$$

gdje su s_k i y_k definirani kao: $s_k := x_{k+1} - x_k$ i $y_k := \nabla(x_{k+1}) - \nabla(x_k)$ (vidi [3]). Pri čemu, u praksi, za početnu aproksimaciju H_1 uzimamo jediničnu matricu I.
DFP ima slična svojstva lokalne konvergencije kao i BFGS, no ona ipak nisu tako dobra u praksi (vidi [4]). Riješit ćemo isti primjer kao i kod BFGS metode, te ćemo usporediti dobivene rezultate.

Primjer 1 *Odredimo minimum funkcije $f : \mathbb{R}^2 \to \mathbb{R}$ definirane s:*

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

Uzmimo istu početnu aproksimaciju kao i kod BFGS; $x^{(0)} = (0.8, 2.7)$. Pokrenimo pripadni MATLAB kod:

```
tic
format long
x0=0.8;
y0=2.7;
syms x y;
```

```
f1(x,y)=(x+2*y-7)^2+(2*x+y-5)^2;
H=eye(2);
J1(x,y)=jacobian([(x+2*y-7)^2+(2*x+y-5)^2], [x, y]);
I=eye(2);
while max(abs(f1(x0,y0)))>eps
    j=J1(x0,y0);
    q=double(H(1,1)/(H(1,2)*H(2,1)+H(2,2)*H(1,1))*(-j(2)+j(1)*...
        H(2,1)/H(1,1)));
    p=double(-j(1)/H(1,1)-H(1,2)/H(1,1)*q);
    x=[x0 y0]
    x1=x0+p;
    y1=y0+q;
    t0=J1(x1,y1)-J1(x0,y0);
    ys=t0(1)*p+t0(2)*q;
    H=1/ys*[p^2 p*q;p*q q^2]+(I-1/ys*[t0(1)*p t0(1)*...
        q; t0(2)*p t0(2)*q])*H*(I-1/ys*[t0(1)*p t0(2)*p; t0(1)*q t0(2)*q]);
    x0=x1;
    y0=y1;
end
toc
```

Dolazimo do istog rješenja $x^{(13)} = (1, 3)$*, no trebalo nam je* 4 *iteracije više nego kod BFGS metode. Također trebalo nam je* 12.48 *sekundi da bismo stigli do rješenja, što je očekivano zbog manje brzine konvergencije DFP metode. To je i jedan od razloga zašto je BFGS metoda puno učestalija u praksi od DFP.*

Literatura

[1] J.E. Dennis Jr., J.J. Moré , *Quasi-Newton methods, motivation and theory*, SIAM Review 19(1977), 46-89str.

[2] J.E. Dennis, R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[3] C.T. Kelley, *Iterative methods for Optimization*, SIAM, Philadelphia, 1999.

[4] R. Scitovski, *Numerička matematika*, Odjel za matematiku, Sveučilište u Osijeku, Osijek, 2000.