# Davidon-Fletcher-Powellov metod za minimizaciju funkcija više nezavisnih promjenjivih uz egzaktnu minimizaciju duž odabranih pravaca

Stefani Kecman

Bakir Bajrovic

## Seminarski rad iz predmeta

## Numerički algoritmi

**Univerzitet u Sarajevu**

**Elektrotehnički fakultet**

**Odsjek za Računarstvo i informatiku**

**2022. godine**

# Sadržaj

# Sažetak

U ovom radu ćemo obraditi temu Davidon-Fletcher-Powellove metode, koja ima primjenu u minimizaciji i optimizaciji brojnih problema. Sam metod će biti obrađen kako sa teoretskog, tako i sa praktičnog aspekta. U teoretskom odjeljku ćemo se baviti uvodnom motivacijom za istraživanje samog metoda, definirati šta znači pronaći minimum i izvršiti minimizaciju. Isto tako, obrazložit ćemo potrebu razvoja pojedinačnih, Quasi-Newton-skih metoda, u koje spada DFP.

Potom će biti prezentiran i objašnjen pseudokod DFP metode, uz prigodno ažuriranje matrice Hessiana i računanje pravca p(x) i tačke x. Na kraju ovog odjeljka će biti prikazana njegova implementacija u programskom jeziku Julia, uzimajući paraboloidnu funkciju za primjer.

Naposlijetku slijedi kratak opis nekoliko praktičnih primjena Davidon-Fletcher-Powellovog algoritma, za rješavanje brojnih fizikalnih problema i njihove optimizacije, među kojima se najviše ističe optimizacija atmosferskog i egzoatmosferskog leta.

Na kraju, kroz zaključak i diskusiju ćemo istaći prednosti i mane ovog metoda (sa praktičnog i teoretskog gledišta), kao i sumirano znanje o njegovoj važnosti i primjeni.

# Abstract

In this work we will revise the subject of Davidon-Fletcher-Powell method, that has its use in minimisation and optimisation of numerous problems. Method itself will be explained from both theoretical and phisical aspect. In theoretical segment we will present introduction to the method and motivation for its study, define what it means to find minimum and do minimisation. Also, we will explain the need for development of Quasi-Newton methods, in which DFP also belongs.

After that, we will present and go through pseudocode for DFP, with convenient updates of Hessian matrix and computing direction p(x) and point X. In the end of this section, implementation of the algorithm in Julia program language will be presented, using paraboloid function for example.

Finally, we will discuss a few practical uses of Davidon-Fletcher-Powell algorithm, for solving numerous physical problems and their optimisations, of which the most standing out one is atmospheric and exoatmospheric flight.

In the end, through conclusion and discussion, we will highlight the pros and cons of this method (from practical and theoretical point of view), as well as sumarise the knowledge of its importance and use.

# Uvod

U ovom odjeljku će se prezentirati uvodna motivacija za istraživanje ovog metoda, uz osnovne informacije o istom te uz kratak opis postupka njegovog izvođenja. Potom slijedi implementacija navedenog algoritma, i to u programskom jeziku Julia, preko kojeg smo demonstrirali njegovo izvršavanje. Vrlo je bitno istaknuti i praktičnu primjenu ovog metoda, što je obrađeno u jednom od poglavlja u nastavku. Naposlijetku, rad je zaokružen diskusijom i zaključkom o dotičnom metodu kao i o njegovim performansama.

DFP metod jedna je od najranijih optimizacija kvazi-Njutnovih metoda, koji je konkretno usko vezan za pronalaženje minimuma funkcija. Prvobitno ga je razvio Davidon (1959. godine), a potom su ga unaprijedili Fletcher i Powell (1963. godine).

DFP algoritam je također poznat kao „algoritam sa varijabilnom metrikom“ [1].

U završnom radu Milinčević Milana, navedeno je da je ovaj metod poznat kao prvi Quasi-Newtonov metod koji zadovoljava jednačinu sekante za multidimenzionalni problem, što trenutno nije tematika ovog rada, ali vrijedi napomenuti [2].

S obzirom da se Davidon-Fletcher-Powellov metod (u narednim navođenjima imena ovog metoda koristit ćemo skraćenicu DFP) koristi za minimizaciju funkcija, podsjetimo se prvo pojma „minimizacija“ (odnoseći se naravno konkretno na minimizaciju funkcija). Pod minimizacijom funkcije više promjenjivih možemo smatrati problem nalaženja vrijednosti $x \in S$ za koju $f(x)$ postiže minimalnu vrijednost (pri čemu je S podskup skupa $R^n$ neka oblast od interesa). Uz pojam minimizacije se usko veže pojam minimuma funkcije, koji može biti globalni ili lokalni. Pod tačkom globalnog (apsolutnog) minimuma funkcije f(x) podrazumijevamo tačku za koju je vrijednost funkcije manja ili jednaka od svake vrijednosti u bilo kojoj drugoj tački skupa S. Ukoliko je ta vrijednost strogo manja, riječ je o strogom globalnom minimumu. Ako posmatramo definiciju minimuma na užoj oblasti okoline neke tačke, koja ne mora obuhvatati čitav skup S, tada govorimo o (strogom) lokalnom (relativnom) minimumu. Razlika između lokalnog i globalnog minimuma je ta što globalni minimum mora biti jedinstven, dok za lokalni to nije slučaj. Problem nalaženja lokalnog minimuma je jednostavniji od istog za globalni, no toj diskusiji ovde nije mjesto [3].

Bitno je još i napomenuti da je minimizacija često usko vezana za problem rješavanja sistema nelinearnih jednačina oblika $f(x) = 0$, jer se taj problem može svesti upravo na problem minimizacije. DFP metod može da se koristi i za jedno i za drugo, međutim u ovom radu naš fokus će biti na minimizaciji. Bitno je naglasiti da je algoritmirana minimizacija funkcija nešto složeniji postupak od klasičnog rješavanja jednačina oblika $f(x) = 0$, s obzirom da ona sa sobom nosi i dopunske uvjete koji moraju biti zadovoljeni u slučaju minimizacije, ali ne i u slučaju rješavanja klasičnih nelinearnih jednačina [2].

Pomenuti dopunski uvjet glasi:

Svaki lokalni minimum $x^*$ kontinualne diferencijabilne funkcije $f$ treba zadovoljiti neophodni uslov $g(x^*) = 0$, pri čemu je $g(x^*) = \nabla f(x^*)$ gradijent funckije u tački minimuma.

Navedena jednačina predstavlja sistem nelinearnih jednačina. Njegovim rješavanjem se dobiva tačka $x^*$. Jedan pristup rješavanju ovog sistema je upravo minimizacija funkcije $f(x^*)$, a taj sistem glasi $g(x) = \begin{bmatrix} \frac{\partial f}{\partial x^{(1)}} \\ \vdots \\ \frac{\partial f}{\partial x^{(n)}} \end{bmatrix}$ [4].

U naslovu teme, vidimo da nije riječ o samo običnoj minimizaciji, nego minimizaciji duž odabranih pravaca, a za takvu minimizaciju pokazali su se kao najuspješniji metodi zasnovani na Newtonovom algoritmu, koje zovemo „Algoritmi za minimizaciju Newtonovog tipa“ [5]

Osnovna ideja koja stoji iza Newtonovog metoda je vršenje iterativne lokalne minimizacije funkcije diferencijalnom funkcijom:

$$x^{(k+1)} = x^{(k)} - F(x^{(k)})^{-1} g^{(k)} \text{ [1]}$$

Jedna mana ovih algoritama je ta što se, u većini njih, u svakoj iteraciji računa Hessian, uz vršenje lijevog matričnog dijeljenja, što poprilično usporava rad algoritama. Da bi se to izbjeglo, predložene su određene modifikacije, te su na osnovu tih modifikacija nastali novi metodi koji se zovu Quasi-Newtonovi metodi (ili metodi promjenjive metrike) [5], a DFP je upravo jedan takav metod. Nakon što smo ustanovili u koju tačno skupinu metoda ga možemo svrstati, te čime se oni bave, ostatak rada će imati fokus samo na naš razmatrani metod.

Quasi-Newtonove metode su upravo i namijenjene rješavanju problema minimizacije za slučajeve kada je klasično računanje matrice Hessiana teško ili gotovo nemoguće.

DFP metodom je pomenuti problem riješen na način da dobivamo aproksimaciju matrice Hessiana kroz iterativni postupak, tačnije aproksimaciju inverzne matrice Hessiana [6].

Nakon što smo se upoznali s nekim osnovnim stavkama i informacijama o algoritmu, vrijeme je da pređemo na konkretniji opis samog rada, postupka izvršavanja, numeričkih performansi, te implementacije algoritma.

# Implementacija algoritma

Slijedi opis implementacije algoritma, te napisan kod u programskom jeziku Julia. Za kreiranje pseudokoda smo se poslužili formulama navedenim u knjizi „Numerički algoritmi“, profesora Jurića.

DFP metod možemo podijeliti na četiri koraka, koje ćemo zvati nulti (inicijalni), prvi, drugi i treći korak.

U nultom, odnosno inicijalnom koraku, biramo pozitivno definitnu matricu $H_0$, za koju se najčešće uzima da je jedinična matrica E, i inicijalnu procjenu $x_0$ vrijednosti $x^*$. Parametar i se postavlja na 1, a on je ustvari broj iteracija koji će se inkrementirati.

U prvom koraku, pomoću iterativnog procesa, tačka x se ažurira preko prethodne vrijedosti, polazeći od tačke $x_0$, sve dok se ne dostigne minimum, ili dok se ne dostigne maksimalan broj iteracija. To se vrši na sljedeći način (isto za svaki metod Newtonovog tipa):

$$x_i = x_{i-1} - \nabla^2 f(x_{i-1}) \backslash \nabla f(x_{i-1})^T$$

Izraz $-\nabla^2 f(x_{i-1}) \backslash \nabla f(x_{i-1})^T$ zapravo predstavlja pravac duž kojeg se vrši minimizacija, koji ćemo nazvati $p$. Kod DFP metoda, kao što smo već naveli, koristi se aproksimacija Hessiana, umjesto njega samog, pa ćemo to i primijeniti u našoj formuli.

Dakle, uzimajući da je $H_i = \nabla^2 f(x_i)^{-1}$, to ćemo sada uvrstiti u našu formulu za pravac:

$$p_i = -H_i \nabla f(x_i)^T$$

Konačno, naša iterativna formula za računanje $x_i$ sada dobija oblik:

$$x_i = x_{i-1} + p_{i-1}$$

Drugi korak postupka jeste ažuriranje naše aproksimacije Hessiana, odnosno $H_i$. Za ovo ažuriranje DFP metod koristi formulu:

$$H_i = H_{i-1} + \frac{\delta_i \delta_i^T}{\gamma_i^T \delta_i} - \frac{H_{i-1} \gamma_i \gamma_i^T H_{i-1}^T}{\gamma_i^T H_{i-1} \gamma_i} ,$$

pri čemu se $\delta_i$ i $\gamma_i$ računaju kao:

$$\delta_i = x_i - x_{i-1}$$

$$\gamma_i = \nabla f(x_i)^T - \nabla f(x_{i-1})^T$$

Treći, posljednji korak, je ustvari samo inkrementiranje brojača i (i++), te vraćanje na prvi korak.

Opisani proces je najbolje vršiti unutar while petlje, koja će prije svake naredne iteracije ispitivati da li je brojač i dostigao maksimalan broj iteracija (koji korisnik zada), ili da li je posljednja izračunata vrijednost x zapravo traženi minimum. Ukoliko je jedan od ova dva

uslova ispunjen, proces će se prekinuti, odnosno daljnje iteracije se neće vršiti. Nakon što smo detaljno opisali sve korake, zabilježene podatke možemo iskoristiti da formiramo pseudokod koji glasi:

*(0) inicijalni korak*

$H_0 \leftarrow E$

i $\leftarrow 1$

**while** $i \neq itmax$

*(1) iterativni korak*

$p_{i-1} \leftarrow -H_{i-1}\nabla f(x_{i-1})^T$

$x_i \leftarrow x_{i-1} + p_{i-1}$

*(2) korak ažuriranja*

$\delta_i \leftarrow x_i - x_{i-1}$

**if** $|\delta_i| \leq \boldsymbol{\varepsilon}$ **or** $\nabla f(x_i)^T \leq \boldsymbol{\varepsilon}$

**break**

$\gamma_i \leftarrow \nabla f(x_i)^T - \nabla f(x_{i-1})^T$

$H_i \leftarrow H_{i-1} + \frac{\delta_i \delta_i{}^T}{\gamma_i{}^T \delta_i} - \frac{H_{i-1}\gamma_i \gamma_i{}^T H_{i-1}{}^T}{\gamma_i{}^T H_{i-1}\gamma_i}$

*(3) korak inkrementiranja*

$i \leftarrow i + 1$

**end while**

Po završetku ovog pseudokoda, traženi minimum je posljednja vrijednost $x_i$, odnosno $x_i$ nakon prekida petlje.

Na osnovu gore navedenog pseudokoga smo sastavili sljedeći programski kod u Juliji:

```
function f(x)
  i = 1
  suma = 0
  while i != length(x)
    suma += 100 * (x[i+1] - x[i]^2)^2 + (x[i] - 1)^2
    i += 1
  end
  return suma
end;

```

```
function f1(x)
    i = 1
    suma = 0
    while i <= length(x)
        suma += x[i] * x[i]
        i += 1
    end
    return suma
end;

using Calculus, LinearAlgebra


i = 1
H = Matrix(1.0I,20,20)

itmax = 1000
x = [1.3, 1.2, 1.3, 1.5, 1.8, 1.2, 1.9, 1.2, 1.1, 1.6, 1.7, 1.4, 1.2, 1.5, 1.5, 1.7, 1.8, 1.9, 1.3, 1.1]
while i != itmax
  p = -H * Calculus.gradient(f1, x)
  x_old = x
  x += p
  delta = x - x_old
  if( norm(delta) <= eps(Float32) ||  norm(Calculus.gradient(f1,x)) <= eps(Float32))
    break
  end
  gama = Calculus.gradient(f1,x) - Calculus.gradient(f1,x_old)
  H_old = H
  H = H_old + delta*transpose(delta) / (transpose(gama)*delta) - H_old*gama*transpose(gama)*transpose(H_old) / ( transpose(gama)*H_old*gama)
  i += 1
end;
println(x)
```

U ovom smo dijelu napisali Rosenbrockovu funkciju. S obzirom da je ona problematična za DFP, pošto je oscilirajuća funkcija, naveli smo i primjer funkcije f1 (paraboloid), za koji bi DFP radio bez problema. Nju smo koristili za demonstraciju algoritma napisanog na osnovu našeg pseudokoda za DFP metod.

# Primjene algoritma u praksi

U ovom odjeljku će biti govora o praktičnoj primjeni DFP metoda.
Davidon – Fletcher- Powellov metod (i uopćeno pitanje minimizacije) ima primjenu primarno u poljima teorije optimizacije, koja je temelj operacionih istraživanja [3].

U dokumentu "The Davidon-Fletcher-Powell Penalty Function Method: A Generalized Iterative Technique For Solving Parameter Optimization Problem" navedeno je da je DFP metoda korištena za rješavanje i primjenu brojnih problema minimizacije uz ograničenja, kao i problema optimizacije parametara. Među njima se ističe rješenje problema orbitnog impulsnog prenosa, u kojem su komponente brzina impulsa i uglovi prenosa.
Također, među pomenutim oblastima primjene se nalazi i oblast optimalnog dizajna nuklearnih reaktora, dizajna optimalnih sistema kontrole zrakoplova, rješavanje sistema jednačina, elektroničkih problema (npr. za radare) te za atmosfersko i egzoatmosfersko optimiziranje leta.

Za svaku osim posljednje dvije primjene, DFP je korišten sa gradijentima ili parcijalnim izvodima. Elektronički problem s primjenom na radare je riješen koristeći račun sa numeričkim, centralnim i gradijentima izračunatim unaprijednim diferencijama.
U slučaju optimizacije atmosferskog i egzoatmosferskog leta također je korišten račun preko numeričkih gradijenata. Pokazalo se da je DFP metod uspješan uz primjenu numeričkih i gradijenata u zatvorenom obliku [7].

DFP metod je uspješno iskorišten za optimizaciju atmosferskog leta koristeći aritmetiku dvostruke preciznosti prilikom pretrage. Pokušaj je bio isto izvesti aritmetikom jednostruke preciznosti, međutim konvergencija se puno lošije postizala. Zaključeno je da je najuspješnija kombinacija jednostruke preciznosti za proračun p(x), zatim dvostruke preciznosti za DFP metod [7].

Kompjuterski program implementacije DFP metode je osmišljen od strane Odjela za matematičku analizu Odjela za razvoj softvera ("Mathematical Analysis Section of the Software Development Branch") NASA-e. U sklopu ovog programa, korisniku je omogućen izbor gradijenata u zatvorenom obliku, gradijenata izračunatih centralnim diferencijama ili gradijenata izračunatih unaprijednim diferencijama. DFP metod jednodimenzionalnom pretragom nudi opcije: kubne interpolacije koristeći nagib u početnoj tački, kubne interpolacije bez nagiba ili metodu zlatnog presjeka. Ovaj program je jednu od svojih primjena pronašao u simulaciji dinamike vozila u svemiru (space vehicle dynamics simulation - SVDS), koji je trenutno u fazi testiranja [7].

Kao što je već navedeno, primarni problemi riješeni Davidon-Fletcher-Powell metodom su problemi optimizacije atmosferskog i egzoatmosferskog leta, riješeni upotrebom parametarske metode korištenjem tehnike DFP funkcije kazne [7].
Sama korištena DFP funkcija kazne je ponekad nazvana programom optimizacije parametara, programom ubrzavajućeg gradijenta ili Davidon programom [7].

Pored navedenih rješenja problema optimizacije, DFP kao i druge Newtonove metode postavljaju temelj razvoju nelinearnog programiranja.

Radovi Khuna i Tuckera na temu potrebnih i dovoljnih uvijeta za optimalno rješenje programskih problema su postavili temelje za istraživanje nelinearnih problema, koji kasnije predstavljaju osnovu istraživanja i razvoja nelinearnog programiranja. Detaljnija obrada ove teme ne pronalazi dovoljnog prostora u ovom odjeljku, ali je vrijedilo navesti [4].

Iako nije nužno vezano uz DFP nego za njegovu roditeljsku metodu, vrijedi napomenuti da je Newtonova metoda je bila poznata u sferi rješenja i pojedinih fizikalnih problema. Cauchy je osmislio prvu primjenu metode najstrmijeg spusta da riješi probleme minimizacije. S duge strane, razvoj simplex metode je primjenjen za rješenje linearnih problema[1].

[1] simplex metod je standardni metod u linearnom programiraju, korišten za rješenje problema optimizacije. S bzirom da to nije tema rada, u detalje nećemo ulaziti

# Zaključak i diskusija

U ovom radu smo istraživali značaj, osobine i primjenu Davidon-Fletcher-Powellove metode minimizacije. Zaključili smo da je DFP, kao i ostale Quasi-Newtonove metode, značajan jer rješava problem pronalaska minimuma za slučajeve kada je inverzna matrica Hessiana gotovo nemoguća za pronaći. Na ovu temu. DFP je vrlo pogodan metod iz razloga iterativnog računanja aproksimacije Hessiana, pri čemu se izbjegava lijevo matrično dijeljenje. Općenito, lijevo matrično dijeljenje bi usporavalo rad algoritma, a DFP je to optimizirao.

Istraživanjem primjene ovog algoritma, pronašli smo podatke o njegovom korištenju u sofisticiranim naučnim branšama. Ističe se optimiziranje atmosferskog i egzoatmosferskog leta, rješavanje fizikalnih problema, u koje spada problem orbitnog impulsnog prenosa. Također se koristi za rješavanje sistema jednačina, optimiziranje sistema kontrole zrakoplova i brojnih drugih problema.

S druge strane, iako je DFP jedan od prvih Quasi-Newtonovih metoda, pokazao se da nije najoptimalniji. Danas se mnogo više koristi njemu sličan Broyden–Fletcher–Goldfarb–Shanno (BFGS) algoritam, koji se također koristi za svrhu minimizacije i optimizacije iz razloga pouzdanosti i numeričke stabilnosti. Pronašli smo podatak da BFGS u prosjeku zahtijeva 10 iteracija da bi postigao preciznost na prvoj decimali, dok je situacija još gora za DFP [8], stoga se BFGS češće koristi. On je umanjio primjenu DFP, te se sada nalazi u većini ovdašnjih softverskih paketa za minimizaciju [5]. Iz ovog razloga, autori seminarskog rada su nailazili na poteškoće pri pretraživanju i pronalasku adekvatnih materijala koji obrađuju ovu materiju, pošto se većina svodi na analizu BFGS, a ne DFP algoritma.

Naposlijetku, napisali smo vlastiti pseudokod, služeći se formulama iz knjige „Numerički algoritmi“, autora prof. Željka Jurića. Programirajući u programskom jeziku Julia smo implementirali ovaj algoritam na primjeru paraboloidne funkcije (f1). Također smo probali izvršiti algoritam na primjeru Rosenbrockove funkcije (f). U slučaju izvršavanja algoritma sa funkcijom f1, dobiju se korektni brojevi. Međutim, u slučaju Rosenbrockove funkcije, f, rješenja su oblika [NaN, NaN, NaN,...,NaN]. Zaključili smo da je riječ o nestabilnosti DFP algoritma, kada su u pitanju oscilirajuće funkcije, a Rosenbrockova funkcija je upravo takva. Dakle, u slučajevima ovakvih funkcija treba izbjegavati ovaj metod.

# Citirani radovi i literatura


[1] W.-T. Chu, *Chapter 11 Quasi-Newton Methods,* 2014.

[2] M. Milinčević, "Quasi-Newtonove metode, završni rad," Sveučilište Josipa Jurja Storssmayera u Osijeku, Odjel za matematiku, Osijek, 2016.

[3] Ž. Jurić, "Numeričko nalaženje minimuma i maksimuma," in *Numerički algoritmi*, Sarajevo, Univerzitet u Sarajevu, 2018, p. 359.

[4] A. M. A. Elsiddieg, "The Davidon Fletcher and Powel Method Tested on Quadratic Functions," *International Journal of Basic Sciences and Applied Computing (IJBSAC),* vol. 1, no. 10, p. 4, 2015.

[5] Ž. Jurić, "Algoritmi za minimizaciju Newtonovog tipa," in *Numerički algoritmi*, Sarajevo, Univerzitet u Sarajevu, 2018, p. 385.

[6] "Remarks on the Davidon-Fletcher-Powell Method," 12 9 2008. [Online]. Available: http://www.math.udel.edu/~angell/Opt/DFP.pdf. [Accessed 25 1 2022].

[7] J. Ivan L. Johnson, "THE DAVIDON-FLETCHER-POWELL PENALTY FUNCTION METHOD: A GENERALIZED ITERATIVE TECHNIQUE FOR SOLVING PARAMETER OPTIMIZATION PROBLEMS," National Aeronautics and Space Administration, Washington, 1976.

[8] M. J. D. Powell, "How bad are the BFGS and DFP methods when the objective function is quadratic?," Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, 1986.