

UNIVERZITET CRNE GORE,

PRIRODNO-MATEMATIČKI FAKULTET

**STEPENA ITERACIJA ZA NALAŽENJE DOMINANTNE SVOJSTVENE VRIJEDNOSTI MATRICE**

SEMINARSKI RAD IZ PREDMETA PARALELNO PROGRAMIRANJE

|  |  |
| --- | --- |
| Profesor:  Dr Igor Jovančević | Studenti:  Đorđe Čabarkapa 5/24 |
| Asistent:  Mr Nikola Pižurica | Miletić Rajan 7/24 |

Podgorica, 2025. godine

**SADRŽAJ**

[1. UVOD 1](#_Toc194415667)

[2. MATEMATIČKE OSNOVE METODE STEPENE ITERACIJE 2](#_Toc194415668)

[3. PRIMJENA STEPENE ITERACIJE ZA PRONALAŽENJE DOMINANTNE SVOJSTVENE VRIJEDNOSTI 4](#_Toc194415669)

[3.1 Analiza stabilnosti sistema 5](#_Toc194415670)

[3.2 Vibraciona analiza konstrukcija 5](#_Toc194415671)

[3.3 PageRank i analiza mreža 5](#_Toc194415672)

[3.4 Smanjenje dimenzionalnosti (PCA) 5](#_Toc194415673)

[3.5 Numerička rješenja parcijalnih diferencijalnih jednačina 5](#_Toc194415674)

[4. SEKVENCIJALNI PRISTUP 5](#_Toc194415675)

[5. PARALELNI PRISTUP 6](#_Toc194415676)

[6. RAZLIKE IZMEĐU SEKVENCIJALNOG I PARALELNOG PRISTUPA 8](#_Toc194415677)

[7. ZAKLJUČAK 9](#_Toc194415678)

[LITERATURA 10](#_Toc194415679)

# UVOD

Problemi pronalaženja svojstvenih vrijednosti predstavljaju temeljni koncept u različitim naučnim i inženjerskim disciplinama, uključujući fiziku (npr. kvantna mehanika), inženjerstvo (analiza struktura, teorija upravljanja), računarstvo (mašinsko učenje, analiza mreža) i matematiku (dinamički sistemi). Na primjer, u inženjerskim analizama, svojstvene vrijednosti i vektori matrice krutosti sistema mogu ukazati na rezonantne frekvencije koje mogu dovesti do ozbiljnih strukturnih kvarova ukoliko se zanemare. U analizi mreža, dominantna svojstvena vrijednost i njen vektor mogu ukazivati na najuticajnije čvorove ili zajednice unutar mreže.

Pojava problema svojstvenih vrijednosti u tako raznovrsnim oblastima podvlači kritičnu potrebu za efikasnim i skalabilnim računarskim tehnikama za analizu svojstava sistema reprezentovanih matricama. Sa rastom složenosti i veličine takvih sistema, rješavanje problema svojstvenih vrijednosti često zahtijeva više procesorske snage nego što pojedinačni procesor može obezbijediti – što ukazuje na neophodnost paralelnog pristupa. Metoda stepene iteracije(poznata kao i Von Misesova iteracija) predstavlja jednostavnu, ali efikasnu iterativnu tehniku za aproksimaciju najveće (po apsolutnoj vrijednosti) svojstvene vrijednosti matrice i njenog pripadajućeg vektora. Tehnički, algoritam funkcioniše kroz uzastopno množenje matrice sa vektorom i normalizaciju rezultata, gdje pod određenim uslovima – kao što su jedinstvena dominantna vrijednost i početni vektor sa nenultom komponentom u njenom pravcu – dolazi do konvergencije (Golub & Van Loan, 2013).

Zbog toga što je osnovna operacija u ovoj metodi upravo množenje matrice i vektora, ona je posebno pogodna za paralelizaciju. Svaki element rezultujućeg vektora može se izračunati nezavisno, što znači da je operacija vrlo laka za distribuciju između više procesora. Ove karakteristike čine metodu stepene iteracije pogodnim izborom za efikasno rješavanje problema dominantnih svojstvenih vrijednosti u sistemima velikih dimenzija na savremenim paralelnim arhitekturama. Savremeni naučni i inženjerski problemi često podrazumijevaju rad sa veoma velikim matricama, kao što su one koje se javljaju u indeksiranju web stranica, analizi društvenih mreža i simulacijama velikih razmjera. Takve matrice premašuju mogućnosti tradicionalnih sekvencijalnih algoritama, kako u vremenskoj složenosti, tako i u memorijskim zahtjevima. Paralelno računanje omogućava raspodjelu zadatka na više procesorskih jedinica, čime se značajno skraćuje vrijeme obrade.

Cilj ovog seminarskog rada jeste detaljno predstavljanje metode stepene iteracija za pronalaženje dominantne svojstvene vrijednosti sa fokusom na njenu paralelizaciju. U radu će biti objašnjen osnovni algoritam, njegove matematičke osnove, realne primjene, sekvencijalni i paralelni pristup kao i prednosti i nedostaci oba pristupa.

# MATEMATIČKE OSNOVE METODE STEPENE ITERACIJE

U ovoj sekciji razmatramo matematičku pozadinu stepene iteracije, njen oslonac na teoriju svojstvenih vrijednosti i vektora, kao i način na koji se objašnjava konvergencija metode prema dominantnoj komponenti.

Neka je A kvadratna matrica dimenzije n x n, koja ima n linearno nezavisnih svojstvenih vektora . Pretpostavlja se da postoji dominantna svojstvena vrijednost tjza svako i da je A dijagonalizabilna. Kvadratna matrica A je dijagonalizabilna ako postoji inverzibilna matrica P takva da je dijagonalna matrica (t.j. van glavne dijagonale ima samo nule). U ovim uslovima, metoda stepene iteracije garantuje konvergenciju ka – svojstvenom vektoru koji odgovara ​. Vektor je svojstveni vektor(engl. *eigenvector*) ako postoji skalar takav da važi:

U ovom kontekstu, λ se naziva svojstvenom vrijednošću (engl. *eigenvalue*), a je odgovarajući svojstveni vektor (Wikipedia – Eigenvalue algorithm, 2025). Svojstveni vektori predstavljaju smjerove u kojima linearna transformacija djeluje isključivo kroz skaliranje. Odnosno, svojstveni vektori su specijalni vektori koje neka matrica (ili linearna transformacija) ne rotira, ne savija, ne mijenja im pravac – već ih samo produžava ili skraćuje, tj. skalira.Dominantna svojstvena vrijednost ​ je ona sa najvećom apsolutnom vrijednošću u spektru matrice. Ova vrijednost je posebno značajna jer u mnogim primjenama upravo ona određuje stabilnost, konvergenciju i dugoročno ponašanje sistema. Međutim, ako dvije ili više svojstvenih vrijednosti imaju istu maksimalnu magnitudu (tj. najveću apsolutnu vrijednost među svojstvenim vrijednostima), konvergencija stepenih iteracija može biti spora ili neodređena.

Stepena iteracija je veoma jednostavan algoritam, ali može konvergirati sporo. Najzahtjevnija operacija u algoritmu je množenje matrice A sa vektorom, pa je efikasan za veoma velike rijetke matrice uz odgovarajuću implementaciju. Brzina konvergencije zavisi od odnosa između najveće( i druge najveće( svojstvene vrijednosti. Precizno, određuje se faktorom:. Što je **manji**, to će metoda **brže konvergirati**, i greška pravca vektora u svakom koraku opada otprilike proporcionalno sa , gdje je broj iteracija (Golub & Van Loan 2013).

Metoda stepene iteracije koristi sledeći rekurzivni izraz:

Ovdje je ​ vektor dobijen u -toj iteraciji, a označava Euklidsku normu. Inicijalni vektor se bira proizvoljno, ali ne smije biti ortogonalan na dominantni svojstveni vektor ​. Početni vektor ​se može uvijek izraziti kao linearna kombinaciju svojstvenih vektora (ako su linearno nezavisni – što jesu kod dijagonalizabilnih matrica) : . Nakon toga on se treba normalizovati: . Zatim se računa , i onda se vrši normalizacija po gore prikazanoj formuli rekurzivnog izraza. Najčešće se koristi **Euklidska (L2) norma**, koja se računa kao:

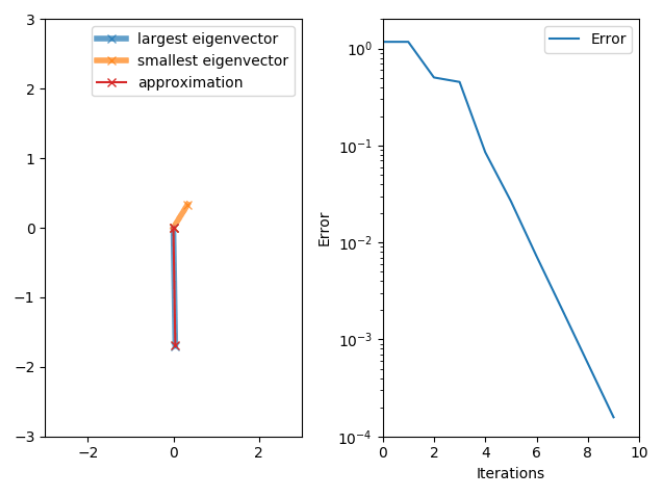
Zatim se provjerava da li je zadovoljen uslov konvergencije: , gdje je **tolerancija greške** – **dozvoljeni prag odstupanja** između dvije uzastopne iteracije vektora. Tolerancija greške se bira proizvoljno, na osnovu: željene tačnosti rezultata, brojčane preciznosti softverskog alata koji se koristi(npr. Python, MATLAB), brzine konvergencije i veličine problema. Vrijednosti poput , ili čak su uobičajene, u zavisnosti od zahtjevane preciznosti. Takođe, preporučuje se postavljanje maksimalnog broja iteracija (npr. 1000) kako bi se spriječila beskonačna petlja u slučaju da se konvergencija ne postigne.

Svakim novim množenjem sa matricom , komponenta u pravcu ​ se pojačava, dok ostale slabe:

Zbog činjenice da jeza sve, člans vremenom dominira, pa

Kada vektor ​ postane dovoljno blizak pravcu svojstvenog vektora odnosno kada se zadovolji uslov konvergencije, odgovarajuća svojstvena vrijednost može se aproksimirati pomoću Rayleigh-ovog kvocijenta:

Ova formula daje vrlo preciznu procjenu ako je .



Slika 1: Vizualizacija algoritma stepene iteracije na 2x2 matrici.

Na slici 1 prikazana je vizualizacija ponašanja metode stepene iteracije. Lijevi grafikon prikazuje matricu sa dva svojstvena vektora, i kako se aproksimacija vektora (crvena oznaka) u toku iteracija postepeno približava pravcu dominantnog svojstvenog vektora (plavi vektor), dok je najmanji svojstveni vektor prikazan narandžasto radi poređenja. Ova grafička ilustracija potvrđuje teorijsko svojstvo metode – vektor aproksimacije se usmjerava ka pravcu ​, što je i cilj postupka. Desni grafikon prikazuje pad greške po iteracijama u logaritamskoj skali. Greška se računa kao || aproksimacija − najveći svojstveni vektor ||. Jasno se vidi eksponencijalni pad, koji potvrđuje da konvergencija metode slijedi obrazac . Već nakon nekoliko iteracija, greška postaje manja od , što pokazuje efikasnost metode u pronalaženju dominantnog svojstvenog vektora.

# PRIMJENA STEPENE ITERACIJE ZA PRONALAŽENJE DOMINANTNE SVOJSTVENE VRIJEDNOSTI

Metoda stepene iteracije predstavlja osnovni i efikasan numerički algoritam za pronalaženje dominantne svojstvene vrijednosti kvadratne matrice, odnosno one svojstvene vrijednosti sa najvećom apsolutnom vrijednošću. U praksi, ova metoda se koristi kada je cilj izdvojiti **najuticajniji modalitet ponašanja sistema** koji je predstavljen matricom – bilo da se radi o fizičkom sistemu, skupu podataka ili matematičkom modelu.

U nastavku će biti prikazane neke od praktičnih primjena stepene iteracije za pronalaženje dominantne svojstvene vrijednosti.

## ****Analiza stabilnosti sistema****

U inženjerskoj praksi, posebno u mehanici i automatizaciji, dominantna svojstvena vrijednost određuje **stabilnost dinamičkih sistema**. Na primjer, u linearnim sistemima oblika , ako je dominantna svojstvena vrijednost realna i negativna, sistem je stabilan; ako je pozitivna, sistem je nestabilan. Stepena iteracija omogućava brzo detektovanje tog ključnog parametra bez potrebe za punom spektralnom dekompozicijom matrice.

## ****Vibraciona analiza konstrukcija****

Kod analiza vibracija u građevinarstvu ili mašinstvu, svojstvene vrijednosti matrice krutosti sistema određuju **prirodne frekvencije** oscilovanja. Prva svojstvena vrijednost često odgovara **najranijem i najsnažnijem modalitetu vibracije**. Stepena iteracija se koristi za izdvajanje upravo tog dominantnog opsega uz značajnu uštedu računarskih resursa.

## ****PageRank i analiza mreža****

U informatici i analizi mreža, dominantna svojstvena vrijednost stohastičke matrice prelaza koristi se za određivanje **relativnog značaja čvorova** u grafovima. Google-ov PageRank algoritam upravo koristi varijaciju stepene iteracije da bi izračunao distribuciju vjerovatnoće posjete web stranicama, što odgovara glavnom svojstvenom vektoru modifikovane matrice web grafa (Gleich, 2004).

## ****Smanjenje dimenzionalnosti (PCA)****

U analizi podataka i mašinskom učenju, metod glavnih komponenti (PCA) koristi svojstvene vrijednosti kovarijacione matrice za identifikaciju **pravaca najveće varijanse u podacima**. Dominantna svojstvena vrijednost pokazuje koliko varijanse u podacima objašnjava prva komponenta, dok pripadajući svojstveni vektor daje pravac te varijanse. Stepena iteracija se koristi za efikasno dobijanje te dominantne komponente, posebno u slučajevima kada je matrica velika i rijetka (Ordonez et al., 2014).

## ****Numerička rješenja parcijalnih diferencijalnih jednačina****

U simulacijama fizičkih procesa, kao što su termalne analize, dinamika fluida i elektromagnetizam, svojstvene vrijednosti matrica koje nastaju diskretizacijom operatora često sadrže važne informacije o ponašanju sistema. Stepena iteracija omogućava fokusiranje samo na najvažniju komponentu bez potrebe za rješavanjem kompletnog spektra.

# SEKVENCIJALNI PRISTUP

U sekvencijalnom izvođenju power iteracije, svi koraci se sprovode na jednom procesorskom jezgru ili u jednom procesu. Ovakav pristup je relativno jednostavan za implementaciju i dobro odgovara problemima srednje veličine. Sekvencijalna verzija metodološki prolazi kroz sljedeće faze (Golub & Van Loan, 2013):

Inicijalizacija: Odabere se početni vektor, najčešće slučajnim izborom ili na osnovu nekog heurističkog kriterijuma. Poželjno je da ovaj vektor nije ortogonalan na traženi svojstveni vektor.

Množenje matrice i vektora: U svakoj iteraciji, postojeći vector množi se sa matricom tj. množenje za gustu matricu od u sekvencijalnom načinu rada zahtijeva oko operacija, pri čemu samo jedna procesorska jedinica izvršava sve aritmetičke zadatke.

Normalizacija: Nakon množenja, dobijeni rezultat y= se normalizuje kako bi se održala numerička stabilnost i spriječilo nekontrolisani rast vrijednosti. Ukoliko koristimo Euklidsku normu , novonastali vektor postaje:

Provjera konvergencije: Posmatra se promjena iz iteracije u iteraciju, na primjer:

ili , gdje su željene granice tolerancije. Ukoliko je jedan od kriterijuma ispunjen, iteracija se prekida.

Računanje svojstvene vrijednosti: Najčešće se najveća svojstvena vrijednost aproksimira kao:

Prednosti sekvencijalnog pristupa uključuju laku implementaciju i relativno malu memorijsku složenost (sve dok je matrica upotrebljiva na jednoj procesorskoj jedinici). Nedostatak je u tome što, za veoma velike matrice, vrijeme izvršavanja može postati izuzetno dugo, jer nema raspodjele računa na više procesora.

# PARALELNI PRISTUP

Klasična metoda stepene iteracije sastoji se od sukcesivnog množenja matrice i vektora, nakon čega slijedi normalizacija rezultujućeg vektora. Kako se radi o operacijama koje zahtijevaju značajna računanja (posebno za velike matrice dimenzija ), jasno je da bi paralelizacija algoritma mogla dovesti do značajnog ubrzanja procesa računanja dominantne svojstvene vrijednosti.

U suštini, paralelizacija algoritma podrazumijeva istovremenu (konkurentnu) raspodjelu poslova između više procesora ili procesorskih jezgara, čime se iskorištavaju savremeni računarski sistemi i postižu velika ubrzanja.

Paralelni pristup metodi stepene iteracije može biti predstavljen u nekoliko jasno definisanih koraka:

Neka je zadana matrica i početni vektor ​. Matrica se dijeli na jednake dijelove između raspoloživih procesorskih jezgara. Pretpostavimo da imamo procesora. Svaki procesor dobija dio matrice koji se sastoji od približno redova. Procesor obrađuje redove od do. Svaki procesor nezavisno računa svoj dio rezultujućeg vektora. Preciznije, za iteraciju svaki procesor ​ računa:

gdje je ​ dio matrice A koji pripada procesoru ​, a ​ je prethodno izračunati (ili početni) vektor. Ovaj korak zahtijeva lokalne proračune na svakom procesoru, bez potrebe za međusobnom komunikacijom među procesorima. (Saad Y., 2011).

Zatim slijedi paralelna normalizacija vektora. Da bismo dobili novi vektor iteracije, neophodno je izvršiti globalnu normalizaciju. Kako je vektor raspodijeljen na procesore, najprije svaki procesor izračunava parcijalnu normu:

Nakon što svaki procesor izračuna svoju parcijalnu normu, slijedi globalna redukcija:

Za ovaj korak neophodna je međusobna komunikacija (npr. MPI\_Allreduce operacija ako koristimo MPI biblioteku). Zatim svaki procesor normalizuje svoj dio vektora lokalno koristeći globalnu normu:

Nakon toga se vrši provjera kriterijuma konvergencije. Za provjeru konvergencije potrebno je porediti novi vektor ​ sa prethodnim ​. Svaki procesor računa lokalnu razliku vektora: . Potom slijedi još jedna globalna redukcija za izračunavanje ukupne razlike:

Ako je ova vrijednost manja od zadate tolerancije , ide se dalje, u slučaju da nije onda se ide u sledeću iteraciju.

Sledeći, ujedno i poslednji korak je aproksimacija dominantne svojstvene vrijednosti (Rayleigh-ev kvocijent). Kada se postigne konvergencija, dominantna svojstvena vrijednost računa se paralelno preko Rayleigh-evog kvocijenta: . Svaki procesor računa parcijalni proizvod a zatim se izvrši globalna redukcija za konačni rezultat svojstvene vrijednosti:

Nakon toga se ponovo provjerava uslov konvergencije, odnosno da li je greška prihvatljiva, po formuli:

Ovdje se se odnosi na to **koliko se svojstvena vrijednost promijenila između iteracija.** Ovaj kriterijum se koristi kada želimo dodatnu sigurnost da je i izračunata svojstvena vrijednost stabilizovana.

Ukoliko greška jeste u opsegu prihvatljive, uzimamo trenutni kao dominantnu vrijednost. U suprotnom, algoritam nastavlja sa narednom iteracijom sve dok ne bude zadovoljen kriterijum konvergencije. (Golub & Van Loan, 2013).

# RAZLIKE IZMEĐU SEKVENCIJALNOG I PARALELNOG PRISTUPA

Vremenska složenost se može uzeti kao najvažnija razlika. Sekvencijalno: Za gustu matricu od , množenje se odvija u operacija po jednoj iteraciji, a izvodi ga samo jedno jezgro. Kod paralelnog pristupa ukupan broj operacija ostaje , ali se dijele između više jezgara ili čvorova, što omogućava znatno ubrzanje, sve dok se troškovi komunikacije ne povećaju do te mjere da naruše dobit od paralelizacije.

Kada je u pitanju memorijska organizacija, pri sekvencijalnom pristupu jedan proces sadrži svu matricu i vector . Ovo može biti problem za ekstremno velike matrice. Dok pri paralelnom svako jezgro čuva samo dio matrice i dio vektora, što omogućava rad sa mnogo većim podacima nego što to može jedna jedinica da prihvati.

Značajna razlika je i složenost implementacije.Sekvencijalni pristup je jednostavniji za kodiranje i razumijevanje. Paralelni pristup zahtijeva upotrebu biblioteka poput MPI, OpenMP ili CUDA. Potrebno je obezbijediti korektnu i efikasnu komunikaciju i sinhronizaciju između čvorova, odnosno procesorskih jezgara.

Za kraj, kao razliku vrijedi pomenuti i praktičnu primjenu.Sekvencijalno riješenje je primjereno za srednje veličine problema i okruženja gdje nema mnogo hardverskih resursa. Dok je paralelno rješenje neophodno za rješavanje vrlo velikih problema (visokodimenzionalne matrice, masivno računanje), gdje se postižu značajne uštede u vremenu.

# ZAKLJUČAK

Paralelna implementacija metode stepene iteracije značajno ubrzava nalaženje dominantne svojstvene vrijednosti za velike matrice. Jasno strukturisani koraci omogućavaju efikasnu implementaciju na više procesorskih jezgara, a ograničenja kao komunikacioni troškovi mogu se optimizovati pažljivim izborom algoritamskih rješenja i tehnologija.

Primjena metode stepene iteracije za pronalaženje dominantne svojstvene vrijednosti matrice je široko rasprostranjena u brojnim disciplinama. Njena jednostavnost, efikasnost i mogućnost lake paralelizacije čine je pogodnom za velike probleme u savremenom računarstvu. Iako metoda ima svoja ograničenja — poput toga što pronalazi samo jednu vrijednost — njena praktična korisnost ostaje velika, naročito kada je potrebno brzo i skalabilno procijeniti dominantne komponente složenog sistema.

U budućnosti se očekuje dalji razvoj paralelnih tehnika koji će omogućiti još veću efikasnost metode stepene iteracije, naročito u aplikacijama poput analize velikih skupova podataka, neuronskih mreža, ili računarskih simulacija na superračunarima.

# LITERATURA

1. **Gleich, D. F. (2004).** PageRank beyond the Web. SIAM Review.
2. **Golub, G. H., & Van Loan, C. F. (2013).** Matrix computations (4th ed.). Johns Hopkins University Press.
3. <https://en.wikipedia.org/wiki/Eigenvalue_algorithm>
4. <https://en.wikipedia.org/wiki/Power_iteration>
5. <https://enastava.matf.bg.ac.rs/~alex/laag/skriptaPT>
6. **Ordonez, C., et al. (2014).** Optimizing the PCA Algorithm. IEEE Transactions on Knowledge and Data Engineering.
7. Saad, Y. (2011). Numerical Methods for Large Eigenvalue Problems. SIAM.