Univerzitet u Nišu
Prirodno-Matematički fakultet

Marko D. Petković

SIMBOLIČKO IZRAČUNAVANJE HANKELOVIH DETERMINANTI I GENERALISANIH INVERZA MATRICA

Doktorska disertacija

Niš, Jun 2008

Mogućnost simboličkog izračunavanja je dragocena za sve matematičke i uopšte, naučne discipline. Uvek kada se traži rezultat izračunavanja koji ne sadrži numeričke greške, simboličko računanje je apsolutno nezamenljiv alat. Zato je razvijen veći broj softvera koji pružaju mogućnosti simboličkog izračunavanja među kojima je najmoćniji i najviše u upotrebi programski paket MATHEMATICA.

U ovoj disertaciji razmatrane su dve neklasične primene simboličkog računanja: simboličko izračunavanje Hankelovih transformacija nizova i simboličko izračunavanje generalisanih inverza konstantnih, racionalnih i polinomijalnih matrica. Disertacija je nastala kao rezultat mog višegodišnjeg bavljenja ovom problematikom što je dovelo do pisanja većeg broja naučnih radova koji su većinom publikovani u eminentnim svetskim časopisima iz ove oblasti.

Sa posebnim zadovoljstvom zahvaljujem se mom mentoru, Prof. Dr Predragu Stanimiroviću, ne samo na pomoći pri izradi ovog rada, već i na velikoj pažnji i vremenu koje mi je posvetio, još od vremena kada mi je kao učeniku gimnazije dao da rešavam prvi naučni problem iz oblasti linearnog programiranja. Od tog vremena, uz njegovu nesebičnu pomoć, prošao sam kroz mnoge oblasti matematike i računarskih nauka što je rezultiralo većim brojem radova koje smo zajedno objavili.

Veliku zahvalnost dugujem i Dr Predragu Rajkoviću kako na nesebičnoj pomoći pri izradi ovog rada, tako i na na stalnoj motivaciji i inspiraciji za naučno-istraživački rad.

Zahvalio bih se i Dr Nebojši Stojkoviću koji je pročitao ovu disertaciju i dao niz korisnih sugestija i čiji su mi prijateljski saveti uvek bili dobrodošli.

Najtoplije se zahvaljujem članovima moje porodice, koji su u granicama svojih mogućnosti takođe doprineli izradi ovog rada.

Sadržaj

Spisak algoritama

Glava 1

Uvod

Simboličko izračunavanje (računanje) predstavlja upotrebu računara za manipulisanje matematičkim izrazima u simboličkoj formi. Koristi se kad je god potrebno dobiti eksplicitan rezultat izračunavanja koji ne sadrži numeričke greške. Zbog toga se simboličko izračunavanje često koristi kod loše uslovljenih problema gde nije moguće numerički precizno izvesti potrebno izračunavanje. Sa druge strane, u mnogim naukama se često sreće potreba za manipulisanjem komplikovanim izrazima koji u sebi sadrže više promenljivih. Najčešće ti izrazi predstavljaju racionalne funkcije kao i polinome od jedne, dve ili više promenljivih. Metodi simboličkog računanja i u ovim slučajevima imaju veliku primenu, a često su i nezamenljivi.

Danas postoji više programskih paketa koji podržavaju simboličko računanje. Oni se nazivaju *softveri za kompjutersku algebru* (CAS, *Computer Algebra Software*), npr. MATHEMATICA, MATLAB, MAPLE, MUPAD, itd. Najpoznatiji i ujedno najmoćniji CAS softver kada je u pitanju simboličko računanje je MATHEMATICA.

Literatura vezana za programski jezik MATHEMATICA je zaista obimna (npr. oficijalne knjige autora Stephena Wolframa [158, 157, 159], pregledni rad [2] kao i sledeće knjige objavljene na srpskom jeziku [123, 75]).

Hankelove determinante imaju veliku primenu u teoriji ortogonalnih polinoma, numeričkoj matematici a takođe i u drugim oblastima matematike i tehničkim naukama. Naročito je važno izračunavanje ovih determinanti u zatvorenom obliku. U skorije vreme publikovan je veći broj naučnih radova u kojima se računaju Hankelove determinante različitih nizova celih brojeva. Objavljen je i veći broj preglednih radova na ovu temu (npr. radovi Krattentnhalera [71, 72]). U bogatoj literaturi vezanoj za Hankelove determinante, postoji veći broj metoda za njihovo izračunavanje. Pomenućemo metod Dodgsonove kondenzacije koji je otkrio C. L. Dodgson i koji je primenljiv u slučaju proizvoljne determinante. Zatim, tu je metod LU faktorizacije [72], i naravno metod baziran na ortogonalnim polinomima i verižnim razlomcima [71, 26]. Metodi bazirani na rezultatima Radouxa i Junoda [110, 111, 112] su novijeg datuma i bazirani su na funkcijama generatrisama polaznog niza momenata.

Generalisani (uopšteni) inverzi matrica predstavljaju uopštenja pojma običnog matričnog inverza. Ako je A regularna kvadratna matrica, tj. ako je $\det A \neq 0$, tada postoji jedinstvena matrica X takva da je $AX = XA = I$, gde je I jedinična matrica. U tom slučaju X je inverzna matrica matrice A i označava se sa A^{-1}. Ukoliko je A singularna matrica (ili pravougaona matrica), tada matrica X sa pomenutim osobinama ne postoji. U tim slučajevima, korisno je odrediti neku vrstu "inverza" matrice A, tj matrice koja će zadržati što je moguće više svojstava inverzne matrice. To je dovelo do pojma *uopštenog inverza* matrice A. Pod uopštenim inverzom matrice A podrazumeva se matrica X koja je u izvesnom smislu pridružena matrici A tako da važi

(1) Uopšteni inverz postoji za klasu matrica koja je šira od klase regularnih matrica (u nekim slučajevima za proizvoljnu matricu A);

(2) Ima neka svojstva običnog inverza;

(3) Svodi se na obični inverz kada je A nesingularna kvadratna matrica.

Ideja o generalisanim inverzima je implicitno sadržana još u radovima C. F. Gaussa iz 1809, i to u vezi sa uvođenjem principa metoda najmanjih kvadrata kod nekonzistentnih sistema. Nakon toga je I. Fredholm, 1903. godine definisao pseudoinverz linearnog integralnog operatora koji nije invertibilan u običnom smislu, a kojim se rešavaju integralne jednačine u slučajevima kada inverzni operator ne postoji. Pokazalo se da tako definisan uopšteni inverzni operator nije jedinstven. W. A. Hurwitz je 1912. godine, koristeći pojam pseudo-rezolvente, opisao čitavu klasu takvih operatora. Generalisani inverzi diferencijalnih operatora implicitno su sadržani u Hilbertovom razmatranju generalisane Greenove funkcije 1904. godine a kasnije su ih proučavali i drugi autori, npr. W. T. Reid 1931., itd.

E. H. Moore [90] je 1920. prvi definisao i proučio jedinstveni generalisani inverz proizvoljne matrice, nazvavši ga "uopštena recipročnost matrice". Moguće je da je do ovih rezultata Moore došao još 1906. godine, mada su prvi rezultati objavljeni tek 1920. godine. Međutim njegov rad malo je bio poznat širokoj javnosti, verovatno zbog specifičnosti terminologije i oznaka. Na primer, ovako je izgledala jedna teorema iz tog rada

Teorema.

$$\mathfrak{U}^C \mathfrak{B}^{1\ II} \mathfrak{B}^{2\ II} \kappa^{12} \cdot) \cdot$$

$$\exists \mid \lambda^{21 \text{ type } \mathfrak{M}_{\kappa^*} \overline{\mathfrak{M}_\kappa}} \ni \cdot S^2 \kappa^{12} \lambda^{21} = \delta^{11}_{\mathfrak{M}_\kappa} \cdot S^1 \lambda^{21} \kappa^{12} = \delta^{22}_{\mathfrak{M}_{\kappa^*}}.$$

Više o Mooreovom rezultatima može se pronaći npr. u [10]. Tek 1955. godine rad R. Penrosea [97] pobudio je pravi interes za izučavanje ove problematike. Penrose je dokazao da je Mooreov inverz zapravo rešenje sistema matričnih jednačina i zbog toga se ovaj inverz danas naziva Moore-Penroseov inverz. Penrose je takođe ukazao na ulogu ovog generalisanog inverza u rešavanju sistema linearnih jednačina.

Teorija, primene i metodi za izračunavanje generalisanih inverza razvijali su se veoma brzo u poslednjih 50 godina. Publikovan je veliki broj naučnih radova i nekoliko monografija, npr. Wang, Wei i Qiao [148], Ben-Izrael i Grevile [10] kao i Rao i Mitra [108]. Poznat je veći broj klasa generalisanih inverza (Moore-Penroseov inverz, Drazinov inverz, grupni inverz, težinski

Moore-Penroseov inverz, $\{i, j, k\}$ inverzi, Bott-Duffinov inverz, itd...). Takođe, predmet intenzivnog proučavanja su, kako generalisani inverzi matrica, tako i generalisani inverzi operatora, elemenata C^* algebri, itd..

Ovde ćemo pomenuti samo neke od metoda za izračunavanje generalisanih inverza matrica. Reprezentacije generalisanih inverza pomoću Jordanove kanoničke forme matrica proučavane su npr. u radovima [16, 32, 38]. Takođe, reprezentacije pomoću faktorizacija potpunog ranga (full rank faktorizacija) date su u radovima [125, 109, 108]. Najpoznatiji rezultat iz ove grupe je rezultat Macduffea [84].

Reprezentacije Moore-Penroseovog i $\{i, j, k\}$ inverza pomoću blok matrica prikazane su u radovima [162, 93, 114]. Žukovski je u svom radu [163] predložio jedan metod koji se zasniva na rekurentnim formulama za rešavanje sistema linearnih jednačina.

Determinantske reprezentacije generalisanih inverza proučavane su u radovima [126, 128, 59]. Reprezentacije generalisanih inverza pomoću graničnih vrednosti date su u radovima [58, 127, 133].

Metode za izračunavanje Drazinovog inverza uveli su Greville [43], Rose [115], Hartwig [51, 52], Campbell, Meyer [16] kao i Wei i Djordjević [151, 153].

Modifikaciju poznatog Leverrierovog metoda (koji je u osnovi metod za računanje koeficijenata karakterističnog polinoma matrice) za računanje široke klase generalisanih inverza (Moore-Penroseovog, Drazinovog, itd...) razvili su Decel [27] (Moore-Penroseov inverz), Grevile [43] i Ji [57] (Drazinov inverz) kao i Stanimirović [124] za široku klasu drugih generalisanih inverza.

Metode pregrađivanja za računanje Moore-Penroseovog inverza, $\{1\}$ inverza kao i težinskog Moore-Penroseovog inverza razvili su Greville [44] kao i Wang i Chen [149].

U skorije vreme proučavani su metodi za računanje generalisanih inverza polinomijalnih i racionalnih matrica. Generalisani inverzi polinomijalnih i racionalnih matrica imaju veliku primenu u automatici i sistemima upravljanja [64, 73]. Modifikaciju Leverrier-Faddevog metoda za polinomijalne i racionalne matrice razvili su Karampetakis [64, 65, 66, 67, 69] kao i Stanimirović [124], Stanimirović i Tasić [134] i Stanimirović i Karampetakis [129]. Takođe i metod pregrađivanja je razvijen za polinomijalne i racionalne matrice [132].

Generalisani inverzi se primenjuju u mnogim oblastima matematike a takođe i u fizici i tehničkim naukama. Simboličko izračunavanje generalisanih inverzna konstantnih, racionalnih i polinomijalnih matrica je značajno zbog primena u tehničkim naukama, naročitu u automatici i sistemima upravljanja.

Ova disertacija predstavlja doprinos simboličkom računanju Hankelovih determinanti i generalisanih inverza matrica. U tu svrhu, predloženi su novi metodi i modifikovani neki postojeći. Većina razmatranih metoda implementirana je u programskom paketu MATHEMATICA. Sve implementacije su besplatne i mogu se preuzeti sa internet adrese:

http://tesla.pmf.ni.ac.yu/people/dexter

Disertacija sadrži rezultate iz različitih matematičkih i oblasti koje pripadaju računarskim naukama: simboličko izračunavanje, teorija algoritama, linearna algebra, numerička matematika, teorija ortogonalnih polinoma, itd...

Disertacija je bazirana na originalnim rezultatima autora koji su publikovani u vodećim međunarodnim časopisima, prvenstveno iz oblasti računarskih nauka [100, 101, 102, 103, 104, 105, 130, 141, 113]. Takođe, sadrži i značajan broj rezultata koji se ovom prilikom prvi put pojavljuju.

Rad je podeljen u 5 glava, svaka glava je podeljena na nekoliko poglavlja a poglavlja na odeljke.

Rezultati sadržani u drugoj glavi odnosiće se na simboličko izračunavanje Hankelovih determinantni, odnosno Hankelovih transformacija nizova. U prvom poglavlju ove glave daćemo prikaz osnovnih definicija i pojmova vezanih za nizove realnih brojeva, sa posebnim osvrtom na nizove celih brojeva.

Drugo poglavlje sadrži pregled poznatih transformacija nizova celih brojeva, uključujući binomnu, invert kao i Hankelovu transformaciju. Biće prikazana i osnovna svojstva ovih transformacija kao i jedna primena Hankelove transformacije u fizici čvrstog stanja.

Osnovni metodi za izračunavanje Hankelove transformacije biće predmet narednog, trećeg poglavlja. To su metod Dodgsonove kondenzacije i Radoux-Junodov metod.

Četvrto poglavlje biće posvećeno teoriji ortogonalnih polinoma. Proučiće se osnovna svojstva ortogonalnih polinoma sa posebnim osvrtom na tročlanu rekurentnu relaciju. Ovo poglavlje predstavlja teorijski uvod za naredno u kome ćemo prezentovati metod za računanje Hankelove transformacije pomoću verižnih razlomaka i ortogonalnih polinoma. Sve etape ovog metoda biće detaljno opisane. Takođe biće prikazan metod za nalaženje težinske funkcije primenom Stieltjesove inverzione formule, kao i metodi za transformaciju težinske funkcije.

U šestom poglavlju druge glave odredićemo Hankelovu transformaciju niza čiji je opšti član jednak sumi dva uzastopna generalisana Catalanova broja. Metod baziran na ortogonalnim polinomima biće primenjen. Ovo poglavlje sadrži originalne rezultate i bazirano je na radu [113].

Predmet proučavanja u sedmom poglavlju je odnos između Hankelove i k-binomnih transformacija. Ove transformacije predstavljaju uopštenje binomne transformacije. Glavni rezultat ovog poglavlja je invarijantnost Hankelove u odnosu na opadajuću binomnu transformaciju. Svi rezultati prikazani u ovom poglavlju su originalni i još uvek neobjavljeni.

U osmom poglavlju druge glave izračunaćemo Hankelovu transformaciju niza generalisanih centralnih trinomnih koeficijenata. Biće korišćen modifikovani metod baziran na ortogonalnim polinomima kao i rezultati iz predhodnog poglavlja. Svi rezultati prikazani u ovom poglavlju su originalni i još uvek neobjavljeni.

Predmet proučavanja poslednjeg, devetog poglavlja ove glave je Hankelova transformacija inverzije niza generalisanih Fibonaccijevih brojeva. Za računanje Hankelove transformacije posmatranog niza koristićemo Radoux-Junodov metod. I ovi rezultati su takođe originalni i još uvek neobjavljeni.

Treća glava ovog rada posvećena je metodima za simboličko računanje generalisanih inverza konstantnih matrica.

U prvom poglavlju ove glave definisaćemo nekoliko klasa uopštenih inverza i proučiti njihova osnovna svojstva. Posebna pažnja biće posvećena Moore-Penroseovom, težinskom Moore-Penroseovom i Drazinovom inverzu.

Drugo poglavlje je posvećeno osnovnim metodima za izračunavanje generalisanih inverza. Detaljno su proučeni metodi bazirani na faktorizacijama potpunog ranga, blokovske reprezentacije i metod Žukovskog.

U trećem poglavlju biće prikazan i detaljno proučen Leverrier-Faddev metod, odnosno modifikacije ovog metoda za računanje Moore-Penroseovog, Drazinovog i široke klase ostalih generalisanih inverza. Svaka varijanta ovog metoda biće predstavljena u obliku algoritma i biće određena odgovarajuća vremenska složenost.

Predmet proučavanja četvrtog poglavlja je metod pregrađivanja. Formulisaćemo tri varijante ovog metoda za računanje Moore-Penroseovog inverza, $\{1\}$ inverza i težinskog Moore-Penroseovog inverza. Takođe biće prikazana vremenska složenost ovog metoda.

Poslednje, peto poglavlje biće posvećeno metodu za računanje $\{i, j, k\}$ inverza u vremenu množenja matrica. Ova složenost je ujedno i teorijski najbolja složenost koju može imati algoritam za računanje generalisanih inverza. Metod je baziran na modifikaciji Courrierovog metoda i generalisanoj Cholesky faktorizaciji. Rezultati prikazani u ovom poglavlju su originalni i još uvek neobjavljeni.

U četvrtoj glavi ovog rada prezentovaćemo metode za računanje generalisanih inverza racionalnih i polinomijalnih matrica. U prvom poglavlju biće date osnovne definicije i svojstva polinomijalnih i racionalnih matrica koja ćemo u nastavku koristiti.

U drugom poglavlju ove glave prikazaćemo interpolacioni metod za računanje Moore-Penroseovog inverza polinomijalnih matrica. Ovaj metod biće baziran na Leverrier-Faddevom metodu. Izračunaćemo vremenske složenosti Leverrier-Faddevog metoda primenjenog na polinomijalne matrice i interpolacionog metoda. Takođe prikazaćemo jedan jednostavan metod za procenu stepena odgovarajućih polinomijalnih matrica. Implementacije ovih algoritama u programskom paketu MATHEMATICA biće testirane na slučajno generisanim test primerima i rezultati testiranja biće prokomentarisani. Svi rezultati ovog poglavlja su originalni i preuzeti iz na v seg rada [130].

Treće poglavlje ove glave biće posvećeno interpolacionom metodu za računanje Drazinovog inverza polinomijalnih matrica bazirano na Leverrier-Faddevom metodu. Kao i u predhodnom, i u ovom poglavlju ćemo konstruisati interpolacioni metod za računanje Drazinovog inverza i izračunati njegovu vremensku složenost. Implementaciju interpolacionog metoda u programskom paketu MATHEMATICA testiraćemo na slučajno generisanim test primerima i rezultati testiranja biće prokomentarisani. Svi rezultati ovog poglavlja su originalni i preuzeti iz našeg rada

[103].

U četvrtom poglavlju konstruisaćemo interpolacioni metod za računanje široke klase generalisanih inverza. Ovaj metod predstavlja uopštenje metoda izloženih u predhodna dva poglavlja. Takođe, konstruisaćemo interpolacione metode za računanje indeksa i ranga polinomijalne matrice. Rezultati izloženi u ovom poglavlju su originalni i preuzeti iz našeg rada [102].

Peto poglavlje je posvećeno modifikaciji metoda pregrađivanja za računanje Moore-Penroseovog inverza racionalnih i polinomijalnih matrica sa dve promenljive. Svi metodi, konstruisani u ovom poglavlju, implementirani su u programskom paketu MATHEMATICA. Rezultati izloženi u ovom poglavlju su originalni i preuzeti iz naših radova [100, 101].

U šestom poglavlju prikazana je modifikacija metoda pregrađivanja za računanje težinskog Moore-Penroseovog inverza racionalnih i polinomijalnih matrica. Definisane su i efektivne strukture kojima se izloženi metodi značajno ubrzavaju. Rezultati izloženi u ovom poglavlju su originalni i preuzeti iz naših radova [141, 104].

Sedmo poglavlje sadži primene teorije generalisanih inverza konstantnih i polinomijalnih matrica u matematičkoj statistici i automatici. Deo vezan za primene u automatici sadrži nekoliko originalnih, još uvek neobjavljenih rezultata.

U petoj, zaključnoj, glavi biće izvršena sistematizacija svih rezultata i biće dato nekoliko predloga za dalja istraživanja.

Glava 2

Simboličko izračunavanje Hankelovih determinanti

U ovoj glavi bavićemo se metodima za simboličko izračunavanje jedne klase determinanti, Hankelovih determinanti. Ova klasa determinanti zapravo predstavlja jednu transformaciju (Hankelovu transformaciju) definisanu na skupu nizova (celih brojeva ili u opštem slučaju realnih ili kompleksnih brojeva). Najpre ćemo opisati nekoliko metoda za računanje Hankelovih determinanti. Ovi metodi će biti iskorišćeni za izračunavanje Hankelove transformacije različitih klasa nizova. Takodje proučićemo i druge transformacije nizova celih brojeva kao i vezu ovih transformacija sa Hankelovom transformacijom.

2.1 Opšti pojmovi vezani za nizove celih i realnih brojeva

Niz realnih brojeva je svaka funkcija

$$a : \mathbb{N}_0 \to \mathbb{R}.$$

Sa $a_n = a(n)$ označićemo opšti član niza a dok ćemo sam niz često obeležavati sa $\{a_n\}_{n\in\mathbb{N}_0}$ (ili $\{a(n)\}_{n\in\mathbb{N}_0}$). Napomenimo da ćemo najviše proučavati nizove celih brojeva (tj. nizove čiji je kodomen skup celih brojeva $\mathbb{Z}$). Napomenimo da mnogi poznati rezultati koje ćemo navoditi, iako su u originalu formulisani samo za nizove celih brojeva, u opštem slučaju važe za proizvoljne realne nizove, pa ćemo ih tako i formulisati.

Nizovi se često zadaju pomoću *funkcije generatrise* (o.g.f, ordinary generating function). Funkcija generatrisa niza $\{a_n\}_{n\in\mathbb{N}_0}$ je funkcija $g(x)$ za koju važi

$$g(x) = \sum_{n=0}^{+\infty} a_n x^n.$$

U nekim slučajevima, razmatraćemo još jedan tip funkcije generatrise koju ćemo nazvati *eksponencijalna funkcija generatrisa* (e.g.f, exponential generating function). Za niz $\{a_n\}_{n\in\mathbb{N}_0}$ eksponencijalna funkcija generatrisa $e(x)$ definisana je sa.

$$e(x) = \sum_{n=0}^{+\infty} \frac{a_n}{n!} x^n.$$

Primetimo da je eksponencijalna funkcija generatrisa niza $\{a_n\}_{n\in\mathbb{N}_0}$ zapravo jednaka funkciji generatrisi niza $\{a_n/n!\}_{n\in\mathbb{N}_0}$. Kažemo da je niz $\{a_n\}_{n\in\mathbb{N}_0}$ *niz momenata* mere μ ako važi

$$a_n = \int_{\mathbb{R}} x^n d\mu(x).$$

Ovakve nizove ćemo često koristiti u nastavku ove glave.

U daljim razmatranjima često ćemo se pozivati na Online enciklopediju celih brojeva (EIS) [121]. Ova enciklopedija je najveća poznata arhiva nizova celih brojeva i sadrži obilje informacija i svojstava za svaki od nizova u njoj (opšti član, funkciju generatrisu, e.g.f, itd...). Autor ove enciklopedije je Neil J. A. Sloane. Na primer, niz Fibonaccijevih brojeva je u ovoj enciklopediji označen sa A000045.

Primer 2.1.1. Niz $\{F(n)\}_{n\in\mathbb{N}_0}$ (A000045) Fibonaccijevih brojeva ima funkciju generatrisu

$$g(x) = \frac{x}{1-x-x^2} = \sum_{k=0}^{\infty} F(n)x^n.$$

Ovu činjenicu je lako dokazati korišćenjem poznate linearne rekurentne jednačine koju zadovoljavaju Fibonaccijevi brojevi

$$F(n) = F(n-1) + F(n-2)$$

sa startnim vrednostima $F(0) = 0$, $F(1) = 1$.

Niz $F(2n+1)$ čiji su prvi članovi $1, 2, 5, 13, 34, 89, \ldots$ ima funkciju generatrisu $\frac{1-x}{1-3x+x^2}$ dok niz $F(2n+2)$ čiji su prvi članovi $1, 3, 8, 21, 55, 144, \ldots$ ima sledeću funkciju generatrisu $\frac{1}{1-3x+x^2}$.

Primer 2.1.2. Niz Catalanovih brojeva (A000108) $\{C(n)\}_{n\in\mathbb{N}_0}$, definisan sa $C(n) = \frac{\binom{2n}{n}}{n+1}$ ima funkciju generatrisu

$$c(x) = \frac{1-\sqrt{1-4x}}{2x}.$$

Ovo je jedan od najproučavanijih i ujedno i najbitnijih nizova celih brojeva. Catalanovi brojevi $C(n)$ predstavljaju niz momenata težinske funkcije

$$\omega(x) = \frac{1}{2\pi}\frac{\sqrt{x(4-x)}}{x}dx$$

na intervalu $[0, 4]$. Prema tome važi

$$C(n) = \frac{1}{2\pi}\int_0^4 x^n \frac{\sqrt{x(4-x)}}{x}dx.$$

Niz isprekidanih Catalanovih brojeva $\{C(\lfloor n/2 \rfloor)(1+(-1)^n)/2\}_{n\in\mathbb{N}_0}$ (prvih nekoliko članova ovog niza su $1, 0, 1, 0, 2, 0, 5, 0, \ldots$) može biti predstavljen kao niz momenata na sledeći način.

$$C(\lfloor n/2 \rfloor)\frac{1+(-1)^n}{2} = \frac{1}{2\pi}\int_{-2}^{2} x^n\sqrt{4-x^2}dx.$$

Da bi dokazali poslednji izraz, dovoljno je da uočimo da je $x^n\sqrt{4-x^2}$ neparna funkcija za neparne vrednosti broja n, pa je njen integral na segmentu $[-2, 2]$ jednak nuli. Za parne vrednosti broja n, $n = 2k$ važi

$$\frac{1}{2\pi}\int_{-2}^{2} x^{2k}\sqrt{4-x^2}dx = 2\int_0^4 y^n\sqrt{4-y}\frac{dy}{\sqrt{y}} = C_k.$$

Primer 2.1.3. Niz centralnih binomnih koeficijenata (A000984) čiji je opšti član $\binom{2n}{n}$ možemo na sličan način da predstavimo kao momente težinske funkcije

$$\binom{2n}{n} = \frac{1}{\pi} \int_0^4 \frac{x^n}{\sqrt{x(4-x)}} dx.$$

Niz isprekidanih centralnih binomnih koeficijenata, $1, 0, 2, 0, 6, 0, 20, 0, \ldots$ čiji su članovi sa parnim indeksom nula a neparni jednaki centralnim binomnim koeficijentima $\binom{2n}{n}$ ima opšti oblik

$$e_n = \binom{n}{\lfloor \frac{n}{2} \rfloor} \frac{1 + (-1)^n}{2}.$$

Eksponencijalna funkcija generatrisa ovog niza je jednaka $J_0(2x)$ gde je $J_0(x)$ Besselova funkcija prve vrste. Drugim rečima važi

$$J_0(2x) = \sum_{k=0}^{\infty} \binom{n}{n/2} \frac{(1 + (-1)^n)}{2n!} x^n.$$

2.2 Transformacije nizova

U ovom poglavlju proučićemo tri najvažnije transformacije brojevnih nizova. Takodje, proučićemo i najvažnija svojstva ovih transformacija.

2.2.1 Binomna i invert transformacija

Definicija 2.2.1. *Binomna transformacija datog niza* $a = \{a_n\}_{n \in \mathbb{N}_0}$ *je niz* $b = \{b_n\}_{n \in \mathbb{N}_0}$ *definisan sa*

$$b_n = \sum_{k=0}^{n} \binom{n}{k} a_k$$

Binomnu transformaciju označavaćemo sa $\mathbf{B}$ *i pisaćemo* $b = \mathbf{B}(a)$.

Ova važna transformacija nizova realnih brojeva je invertibilna. Sledeća lema opisuje inverznu transformaciju $\mathbf{B}^{-1}$ binomnoj transformaciji.

Lema 2.2.1. *Binomna transformacija je invertibilna. Ako važi* $b = \mathbf{B}(a)$ *onda je*

$$a_n = \sum_{k=0}^{n} (-1)^{n-k} \binom{n}{k} b_k.$$

Ukoliko koristimo funkciju generatrisu za opisivanje niza, tada je binomna transformacija niza čija je funkcija generatrisa $g(x)$, niz čija je funkcija generatrisa definisana sa $\frac{1}{1-x} g(\frac{x}{1-x})$. Slično, ukoliko niz opisujemo eksponencijalnom funkcijom generatrisom $e(x)$, tada je binomna transformacija ovog niza niz čija je eksponencijalna funkcija generatrisa jednaka $\exp(x)e(x)$.

Primer 2.2.1. Ukoliko dvaput primenimo binomnu transformaciju na niz $\{e_n\}_{n \in \mathbb{N}_0}$ isprekidanih centralnih binomnih koeficijenata dobijamo niz centralnih binomnih koeficijenata $(1, 2, 6, 20, 70, \ldots)$

$\{\binom{2n}{n}\}_{n\in\mathbb{N}_0}$. Polazni niz $\{e_n\}_{n\in\mathbb{N}_0}$ ima funkciju generatrisu $\frac{1}{\sqrt{1-4x^2}}$. Primenjujući binomnu transformaciju na ovu funkciju generatrisu dobijamo

$$\frac{1}{1-x}\frac{1}{\sqrt{1-4(\frac{1}{1-x})^2}}=\frac{1}{\sqrt{1-2x-3x^2}}.$$

Ova funkcija predstavlja funkciju generatrisu niza centralnih trinomnih koeficijenata A002426 o kome će biti reči na kraju ove glave. Primenjujući još jednom binomnu transformaciju dobijamo sledeću funkciju generatrisu

$$\frac{1}{1-x}\frac{1}{\sqrt{1-2\frac{x}{1-x}-3(\frac{x}{1-x})^2}}=\frac{1}{\sqrt{1-4x}}.$$

Ovo je zapravo funkcija generatrisa niza centralnih binomnih koeficijenata. Ukoliko sada posmatramo eksponencijalne funkcije generatrise možemo da zaključimo da niz $\binom{2n}{n}$ ima eksponencijalnu funkciju generatrisu jednaku $\exp(2x)J_0(2x)$.

Sada definišemo invert transformaciju. Ova transformacija je definisana samo za nizove $\{a_n\}_{n\in\mathbb{N}_0}$ za koje važi $a_0=0$ i $a_1=1$.

Definicija 2.2.2. *Neka je $\{a_n\}_{n\in\mathbb{N}_0}$ niz koji zadovoljava uslove $a_0=0$ i $a_1=1$ i neka je $f(x)$ funkcija generatrisa ovog niza. Invert transformacija niza $\{a_n\}_{n\in\mathbb{N}_0}$ je niz $\{b_n\}_{n\in\mathbb{N}_0}$ čija je funkcija generatrisa $g(x)$ definisana na sledeći način*

$$g(x)=\frac{f(x)}{1-f(x)}.$$

Invert transformaciju ćemo označavati sa **INV** *i pisaćemo $b=\mathbf{INV}(a)$.*

2.2.2 Hankelova transformacija

Definisaćemo Hankelovu transformaciju koju ćemo proučavati u nastavku ove glave.

Definicija 2.2.3. Hankelova transformacija *datog niza $a=\{a_n\}_{n\in\mathbb{N}_0}$ je niz $\{h_n\}_{n\in\mathbb{N}_0}$ determinanti Hankelovih matrica $H_n=[a_{i+j-2}]_{i,j=1}^n$, tj.*

$$a=\{a_n\}_{n\in\mathbb{N}_0}\quad\stackrel{\mathbf{H}}{\Longrightarrow}\quad h=\{h_n\}_{n\in\mathbb{N}_0}:\quad h_n=\det\begin{bmatrix} a_0 & a_1 & \cdots & a_n \\ a_1 & a_2 & & a_{n+1} \\ \vdots & & \ddots & \\ a_n & a_{n+1} & & a_{2n}\end{bmatrix} \tag{2.1}$$

Hankelovu transformaciju označavaćemo sa **H** *i pisaćemo $h=\mathbf{H}(a)$.*

Hankelove determinante se još nazivaju i *persimetrične* ili *Turanove* determinante. Iako se determinante Hankelovih matrica proučavaju već duže vreme, termin Hankelova transformacija uveo je Layman 2001 godine u radu [77]. U istom radu, Layman je dokazao da je Hankelova transformacija invarijantna u odnosu na binomnu i invert transformaciju.

Teorema 2.2.2. **[77] (Layman 2001)** *Hankelova transformacija* $\mathbf{H}$ *je invarijantna u odnosu na binomnu transformaciju* $\mathbf{B}$ *kao i invert transformaciju* $\mathbf{INV}$*, tj. važi* $\mathbf{H}(\mathbf{B}(a)) = \mathbf{H}(a)$ *i* $\mathbf{H}(\mathbf{INV}(a)) = \mathbf{H}(a)$ *za svaki niz* $a = \{a_n\}_{n\in\mathbb{N}_0}$.

Primetimo da je činjenica da Hankelova transformacija $\mathbf{H}$ nije invertibilna, sledi kao direktna posledica predhodne teoreme.

Postoji više različitih metoda za računanje Hankelovih determinanti. Metod Dodgsonove kondenzacije[29, 71] može se direktno primeniti na računanje Hankelovih determinanti (ovaj metod je primenljiv na bilo koje determinante). Ista situacija je i sa metodom LU dekompozicije. Metod koji su opisali Eğecioğlu, Redmond i Ryavec [31] povezuje računanje Hankelovih determinanti sa rešavanjem diferencijalnih konvolucionih jednačina i može se primeniti u nekim slučajevima gde drugi metodi ne mogu.

Najpopularniji metod za računanje Hankelovih determinanti je metod baziran na verižnim razlomcima odnosno ortogonalnim polinomima. Mi ćemo detaljnije proučiti ovaj metod u naredna dva poglavlja.

Takodje u radovima [31, 71, 72, 96, 110, 111, 112] izračunata je Hankelova transformacija velikog broja različitih klasa nizova. U narednom primeru razmotrićemo dva takva izračunavanja.

Primer 2.2.2. Hankelova transformacija niza Catalanovih brojeva $\{C_n\}_{n\in\mathbb{N}_0}$ je niz $\{1\}_{n\in\mathbb{N}_0}$. Drugim rečima svaka determinanta

$$|1|, \qquad \begin{vmatrix} 1 & 1 \\ 1 & 2 \end{vmatrix}, \qquad \begin{vmatrix} 1 & 1 & 2 \\ 1 & 2 & 5 \\ 2 & 5 & 14 \end{vmatrix}, \qquad \dots$$

ima vrednost 1.

Hankelova transformacija niza centralnih binomnih koeficijenata je niz $\{2^n\}_{n\in\mathbb{N}_0}$. Drugim rečima važi

$$|1| = 1, \qquad \begin{vmatrix} 1 & 2 \\ 6 & 20 \end{vmatrix} = 2, \qquad \begin{vmatrix} 1 & 2 & 6 \\ 2 & 6 & 20 \\ 20 & 70 & 252 \end{vmatrix} = 4, \qquad \dots$$

2.2.3 Primena Hankelove transformacije u fizici čvrstog stanja

Postoji mnogo primena Hankelove transformacije (Hankelovih determinanti) u kombinatorici. Primene u prebrojavanju Astečkih dijamanata mogu da se nadju, na primer, u [12, 53]. Takodje, primena koja se tiče square-ice modela data je u radu [20]. Veza izmedju Hankelovih determinanti i sistema kompjuterske algebre je uspostavljena u [118]. Takodje je dat jedan algoritam za računanje najvećeg zajedničkog delioca polinoma zasnovan na Hankelovim matricama.

Ovde ćemo razmotriti jednu primenu u fizici čvrstog stanja. Enrico Fermi, John Pasta i Stanislaw Ulam su 1955. godine u Los Alamosu izveli naizgled bezazlen kompjuterski eksperiment. Oni su razmatrali prost model nelinearnog jednodimenzionog kristala opisujući lanac čestica koje interaguju samo sa najbližim susedima.

Označimo sa $q_n = q_n(t)$ pomeraj n-te čestice iz svog ravnotežnog položaja, sa $p_n = p_n(t)$ njen moment (masa čestice je $m = 1$), a sa $V(r)$ potencijal interakcije. Hamiltonian sistema može da se napiše kao

$$\mathcal{H}(p,q) = \sum_{n\in\mathbb{Z}} \left(\frac{p_n^2}{2} + V(q_{n+1} - q_n)\right)$$

Jednačine kretanja su

$$\begin{aligned} p_n' &= -\frac{\partial \mathcal{H}}{\partial q_n} = V(q_n - q_{n-1}) - V(q_{n+1} - q_n), \\ q_n' &= \frac{\partial \mathcal{H}}{\partial p_n} = p_n. \end{aligned}$$

Eliminacijom p_n dobijamo sledeći beskonačni sistem diferencijalnih jednačina drugog reda

$$q_n'' = V'(q_n - q_{n-1}) - V'(q_{n+1} - q_n). \tag{2.2}$$

Fermi, Pasta i Ulam su razmatrali slučaj konačno mnogo čestica (ovaj slučaj se dobija odredjivanjem periodičnih graničnih uslova) i harmonijsku interakciju $V(r) = r^2/2$. Rešenje je tada dato superpozicijom pridruženih normalnih modova. Bilo je očekivano da će za neke početne uslove, posle dovoljno dugog vremena, energija biti uniformno rasporedjena po svim normalnim modovima. Medjutim, na opšte iznenadjenje, eksperiment je pokazao kvazi-periodično kretanje sistema, umesto očekivane ravnomerne raspodele energije (termodinamičke ravnoteže).

Posle toga, glavna pažnja je bila usmerena na to da se nadje potencijal $V(r)$ za koji gornji sistem ima soliton rešenja. Razmatrajući adicione formule za eliptičke funkcije, Morikazu Toda je došao do potencijala $V(r) = e^{-r} + r - 1$. Zamenom ovog potencijala u (2.2) dobijamo jednačinu koja je sada poznata kao *Toda jednačina*

$$q_n'' = e^{q_{n-1}-q_n} - e^{q_n - q_{n+1}}. \tag{2.3}$$

Uvodeći smenu promenljivih $\tau_n = \log(q_{n+1}/q_n)$, Toda jednačina (2.3) može da se svede na bilinearnu jednačinu

$$\tau_n''\tau_n - (\tau_n')^2 = \tau_{n+1}\tau_{n-1}. \tag{2.4}$$

Za semi-beskonačnu rešetku, sa graničnim uslovima $\tau_{-1} = 0$ i $\tau_0 = 1$, rešenje je dato Hankelovom determinantom

$$\tau_n = \det[a_{i+j-2}]_{i,j=1,\dots,n}, \quad a_0 = \tau_1,\ a_i = a_{i-1}', \quad i,n \in \mathbb{N}.$$

Takodje, postoje i neki drugi slučajevi kada rešenje jednačine (2.3) može da bude izraženo koristeći Hankelove determinante, ili drugi tip determinanti. Više o Toda jednačini i njenim rešenjima može da se nadje, na primer, u [62, 63, 89]. U zadnje vreme, Toda jednačinama se obraća posebna pažnja zbog njihove uloge u vezi izmedju teorija kvantne gravitacije i teorije solitona.

2.3 Osnovni metodi za računanje Hankelove transformacije

U ovom poglavlju izložićemo dva osnovna metoda za računanje Hankelove transformacije. Metod baziran na verižnim razlomcima, odnosno ortogonalnim polinomima biće predmet razmatranja u naredna dva poglavlja.

2.3.1 Metod Dodgsonove kondenzacije

Ovaj metod omogućava efikasan i kratak induktivni dokaz pretpostavljenog rešenja determinante. Jedina poteškoća je što se odgovarajuće rešenje mora intuitivno naslutiti. Ovaj metod se često vezuje za Charlesa Ludwiga Dodgsona, poznatijeg kao Lewis Carroll. Ipak, identitet na koji se bazira ovaj metod najverovatnije potiče od P. Desnanota.

Teorema 2.3.1. **[71] (Dodgsonova kondenzacija)** *Neka je A data matrica formata $n \times n$. Označimo sa $A_{-i_1,\ldots,i_k}^{-j_1,\ldots,j_k}$ podmatricu matrice A koja se dobija izbacivanjem vrsta $i_1,\ldots,i_k$ i kolona $j_1,\ldots,j_k$ iz matrice A. U tom slučaju važi*

$$\det A \cdot \det A_{-1,n}^{-1,n} = \det A_{-1}^{-1} \cdot \det A_{-n}^{-n} - \det A_{-1}^{-n} \cdot \det A_{-n}^{-1}.$$

Poslednja teorema važi za proizvoljnu kvadratnu matricu. Nekoliko različitih izračunavanja determinanti korišćenjem metoda Dodgsonove kondenzacije dato je u radu [71]. Sada ćemo formulisati ovaj metod za Hankelove determinante.

Posledica 2.3.2. *Neka je $a = \{a_n\}_{n\in\mathbb{N}_0}$ proizvoljan niz i neka je $a' = \{a_{n+1}\}_{n\in\mathbb{N}_0}$ i $a'' = \{a_{n+2}\}_{n\in\mathbb{N}_0}$. Sa h, h' i h'' označimo Hankelovu transformaciju nizova a, a' i a''. Tada važi*

$$h_n h''_{n-2} = h''_{n-1} h_{n-1} - (h'_{n-1})^2.$$

2.3.2 Radoux-Junodov metod baziran na funkcijama generatrisama

Ovaj metod je proistekao iz rezultata do kojih su došli Radoux 2000. godine kao i Junod 2003. godine. Ovi rezultati povezuju funkcije generatrise (klasičnu i eksponencijalnu) sa izračunavanjem Hankelove transformacije.

Rezultati do kojih je došao C. Radoux u radovima [110, 111, 112] impliciraju sledeću lemu.

Lema 2.3.3. **[61, 110] (Radoux 2000)** *Označimo sa $F(z) = \sum_{n\in\mathbb{N}_0} a_n \frac{z^n}{n!}$, e.g.f niza $\{a_n\}_{n\in\mathbb{N}_0}$. Ako postoji niz funkcija $\{F_n(z)\}_{n\in\mathbb{N}_0}$ i niz brojeva $\{d_n\}_{n\in\mathbb{N}_0}$ takvi da važi*

(1) $F_k^{(n)}(0) = 0$ *za svako* $n < k$,

(2) $\displaystyle\sum_{k=0}^{+\infty} d_k F_k(y) F_k(z) = F(y+z)$,

tada je Hankelova transformacija niza $\{a_n\}_{n\in\mathbb{N}_0}$ jednaka $h_n = \prod_{k=0}^{n} d_k (F_k^{(k)}(0))^2$.

Korišćenjem predhodne leme, sledeće dve teoreme mogu da se dokažu. Do ovih rezultata došao je Junod u radu [61]. Prva teorema daje izraz za Hankelovu transformaciju nizova čija e.g.f. ima odredjena svojstva.

Teorema 2.3.4. [61] (Junod 2003) *Neka je* $\{\mu_n\}_{n\in\mathbb{N}_0}$ *niz čija je e.g.f. jednaka* $F(x)$. *Pretpostavimo da važi* $F(z) = e^{G(z)}$ *gde je* $G(z)$ *diferencijabilna funkcija takva da je* $G(0) = 0$ *i* $g(z) = G'(z) - G'(0)$ *zadovoljava* $g'(z) = \alpha + \beta g(z) + \gamma g(z)^2$ *za neke vrednosti parametara* $\alpha, \beta, \gamma \in \mathbb{R}$, $\alpha \neq 0$. *Tada je*

$$F(y+z) = \sum_{k=0}^{+\infty} \frac{(1+\gamma)(1+2\gamma)\cdots(1+(k-1)\gamma)}{k!\alpha^k} g(y)^k F(y) g(z)^k F(z).$$

Hankelova transformacija niza $\{\mu_n\}_{n\in\mathbb{N}_0}$ *jednaka je*

$$h_n = \alpha^{\binom{n+1}{2}} \prod_{k=0}^{n} \left(k!(1+\gamma)(1+2\gamma)\cdots(1+(k-1)\gamma)\right).$$

Druga teorema daje izraz za računanje Hankelove transformacije nizova čija o.g.f. ima odredjena svojstva. Pre nego što formulišemo ovu teoremu, uvešćemo operator ∇ na sledeći način

$$\nabla f(z) = \frac{f(z) - f(0)}{z}.$$

Teorema 2.3.5. [61] (Junod 2003) *Neka je dat niz* $\{\mu_n\}_{n\in\mathbb{N}_0}$ *čija je funkcija generatrisa* $F(x)$. *Pretpostavimo da važi* $F(z) = \frac{1}{1-G(z)}$ *gde je* $G(z)$ *funkcija tako da je* $G(0) = 0$. *Pretpostavimo dalje da funkcija*

$$g(z) = \nabla G(z) - \nabla G(0) = \frac{G(z)}{z} - G'(0)$$

zadovoljava $g(z) = z(\alpha + \beta g(z) + \gamma g(z)^2)$ *za neke vrednosti parametara* $\alpha, \beta, \gamma \in \mathbb{R}$, $\alpha \neq 0$. *Tada je Hankelova transformacija niza* $\{\mu_n\}_{n\in\mathbb{N}_0}$ *jednaka*

$$h_n = \alpha^{\binom{n+1}{2}} \gamma^{\binom{n}{2}}.$$

Poslednje dve teoreme mogu biti upotrebljene za računanje Hankelove transformacije različitih nizova. U radu [61] ove dve teoreme su primenjene na računanje Hankelove transformacije niza Hermiteovih polinoma, Bellovih polinoma, Ojlerovih brojeva, itd...

2.4 Ortogonalni polinomi

U sledećem poglavlju proučavamo metod za računanje Hankelove transformacije baziran na ortogonalnim polinomima. Ovaj metod je posledica činjenice da su ortogonalni polinomi i Hankelove determinante veoma usko povezani. Zato ćemo u ovom poglavlju da napravimo jedan kratak pregled teorije ortogonalnih polinoma. Uvešćemo definicije i formulisati tvrdjenja koja ćemo kasnije koristiti. Više o teoriji ortogonalnih polinoma može da se pronadje u veoma bogatoj literaturi koja tretira ovu oblast. Primeri su poznate monografije [17, 40, 140, 145, 146].

2.4.1 Definicija i osnovna svojstva

Neka je $\lambda : \mathbb{R} \to \mathbb{R}$ neopadajuća funkcija i pretpostavimo da postoje konačne granične vrednosti $\lim_{t\to\pm\infty} \lambda(t)$. Ova funkcija indukuje pozitivnu meru $d\lambda$ na skupu $\mathbb{R}$. Takodje je sledećim izrazom

$$\mathcal{L}(f) = \int_{\mathbb{R}} f d\lambda, \qquad (f \in \mathcal{C}(\mathbb{R})) \tag{2.5}$$

definisan pozitivni linearni funkcional $\mathcal{L}$ na skupu svih neprekidnih realnih funkcija jedne promenljive $\mathcal{C}(\mathbb{R})$. Štaviše, na osnovu Rieszove reprezentacione teoreme, za svaki pozitivni funkcional $\mathcal{L}$ na $\mathcal{C}(\mathbb{R})$ postoji pozitivna mera $d\lambda$ takva da jednačna (2.5) važi.

Pretpostavimo da su svi momenti mere $d\lambda$ konačni, tj. da je

$$\mu_n = \int_{\mathbb{R}} x^n d\lambda \in \mathbb{R}, \qquad (n \in \mathbb{N}_0).$$

Za svaki par polinoma (i uopšte neprekidnih funkcija) $p, q \in \mathbb{R}[x]$ možemo definisati skalarni proizvod na sledeći način

$$(p, q) = \int_{\mathbb{R}} p(x)q(x)d\lambda.$$

Ovaj skalarni proizvod indukuje sledeću normu

$$\|p\| = (p, p)^{1/2} = \left(\int_{\mathbb{R}} p(x)^2 d\lambda\right)^{1/2}.$$

Sada ćemo uspostaviti vezu izmedju Hankelove transformacije i predhodno definisanog skalarnog proizvoda. Neka je $h = \mathbf{H}(\mu)$, i neka je $\{H_n\}_{n\in\mathbb{N}_0}$ niz Hankelovih matrica $H_n = [\mu_{i+j}]_{0\le i,j\le n-1}$. Za svaki polinom $p(x) = p_n x^n + p_{n-1}x^{n-1} + \ldots + p_1 x + p_0$ označimo sa $\mathbf{p} = \begin{bmatrix} p_0 & p_1 & \cdots & p_{n-1} \end{bmatrix}^T$ vektor njegovih koeficijenata. Tada važi $(p, q) = \mathbf{p}^T H_n \mathbf{q}$ za svaka dva polinoma $p, q \in \mathbb{R}[x]$ stepena n. Relacija takodje važi i ako su stepeni polinoma p i q različiti, ukoliko vektor $\mathbf{p}$ ili $\mathbf{q}$ na odgovarajući način dopunimo nulama.

Propozicija 2.4.1. *Skalarni proizvod* $(\cdot, \cdot)$ *je pozitivno definitan na* $\mathbb{R}[x]$ *ako i samo ako važi* $h_n > 0$ *za svako* $n \in \mathbb{N}_0$.

Uslov Propozicije 2.4.1 je zadovoljen ako funkcija λ ima beskonačno mnogo tačaka rasta. Rećićemo da je tačka t_0 *tačka rasta* funkcije $\lambda(t)$ ako važi $\lambda(t_0 - \epsilon) < \lambda(t_0 + \epsilon)$ za svako $\epsilon > 0$. Skup svih tačaka rasta funkcije λ se naziva *nosač* mere $d\lambda$ i označava sa $\mathrm{supp}(d\lambda)$.

Definicija 2.4.1. *Monični polinomi* $\pi_k(t) = t^k + \ldots \in \mathbb{R}[x]$, $k = 0, 1, \ldots$ *se nazivaju monični ortogonalni polinomi u odnosu na meru* $d\lambda$ *ako važi*

$$\begin{aligned} (\pi_k, \pi_l) &= 0, \qquad k \neq l, \quad (k, l \in \mathbb{N}_0) \\ \|\pi_k\| &> 0, \qquad (k \in \mathbb{N}_0). \end{aligned}$$

Može se dokazati da je Propozicija 2.4.1 potreban i dovoljan uslov postojanja niza moničnih ortogonalnih polinoma $\{\pi_n\}_{n\in\mathbb{N}_0}$. Prema tome, u ostatku ovog poglavlja pretpostavićemo da je ovaj uslov ispunjen. Trivijalno važi $\pi_0(x) = 1$, pošto su svi polinomi π_n monični.

Lema 2.4.2. *Skup polinoma* $\{\pi_0, \pi_1, \dots, \pi_n\}$ *je linearno nezavisan. Štaviše, svaki polinom* $p \in \mathbb{R}[x]$ *stepena najviše* n *može se na jedinstven način predstaviti u obliku*

$$p = \sum_{k=0}^{n} c_k p_k \tag{2.6}$$

za neke konstante c_k*.*

Drugim rečima, predhodna lema tvrdi da ortogonalni polinomi $\pi_0, \pi_1, \dots, \pi_n$ formiraju bazu podprostora svih polinoma koji su stepena najviše n. Primetimo da niz polinoma $\{\pi_n\}_{n\in\mathbb{N}_0}$ možemo dobiti primenom Gram-Schmidtovog metoda ortogonalizacije na niz $\{x^n\}_{n\in\mathbb{N}_0}$. Prema tome, usvajamo da je $\pi_0 = 1$ i rekurzivno računamo ostale članove niza moničnih ortogonalnih polinoma

$$\pi_k = x^k - \sum_{i=0}^{k-1} c_i \pi_i, \quad c_i = \frac{(x^k, \pi_i)}{(\pi_i, \pi_i)}.$$

Kako je skalarni proizvod $(\cdot, \cdot)$ pozitivno definitan, polinom π_k je jednoznačno definisan i ortogonalan na sve ostale polinome π_j, $j \neq k$.

Deljenjem svakog polinoma π_n sa njegovom normom $\|\pi_n\|$ dobijamo niz *ortonormiranih polinoma* $\{\pi_n(x)/\|\pi_n\|\}_{n\in\mathbb{N}_0}$ koji su takodje ortogonalni na skalarni proizvod $(\cdot, \cdot)$ i čija je norma jednaka 1.

Kažemo da je mera $d\lambda$ apsolutno neprekidna ako je $d\lambda = w(t)dt$ gde je $w(x)$ nenegativna integrabilna funkcija na $\mathbb{R}$ koju nazivamo *težinska funkcija (težina)*. Formalnije, kažemo da je mera $d\lambda$ apsolutno neprekidna (u odnosu na Lebesgueovu meru) ako postoji Lebesgue merljiva funkcija $w(x)$, tako da važi $d\lambda(S) = \int_S w(x)dx$ za svaki (Lebesgue) merljivi skup S. Može se dokazati da je $\operatorname{supp}(d\lambda) = \{x \in \mathbb{R} \mid w(x) \neq 0\}$. Zato ćemo pojam nosača prirodno proširiti i na težinske funkcije i pisaćemo $\operatorname{supp}(w) = \operatorname{supp}(d\lambda)$.

Za odgovarajuće ortogonalne polinome $\{\pi_n\}_{n\in\mathbb{N}_0}$ kažemo da su ortogonalni u odnosu na težinu $w(x)$ (odnosno da su pridruženi težini $w(x)$).

Sada ćemo dati prvu vezu izmedju Hankelovih determianti i ortogonalnih polinoma koja omogućava eksplicitnu reprezentaciju opšteg člana niza ortogonalnih polinoma u funkciji niza momenata $\{\mu_n\}_{n\in\mathbb{N}_0}$.

Teorema 2.4.3. *Polinomi* $\pi_n(x)$ *mogu se predstaviti u sledećem obliku*

$$\pi_n(x) = \frac{1}{h_{n-1}} \det \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_n \\ \mu_1 & \mu_2 & & \mu_{n+1} \\ \vdots & & \ddots & \\ \mu_{n-1} & \mu_n & \cdots & \mu_{2n-1} \\ 1 & x & \cdots & x^n \end{bmatrix}$$

gde je $h = \mathbf{H}(\mu)$ *Hankelova transformacija niza momenata* $\mu = \{\mu_n\}_{n\in\mathbb{N}_0}$*.*

2.4.2 Tročlana rekurentna relacija

Tročlana rekurentna relacija koju zadovoljavaju ortogonalni polinomi predstavlja jedno od najvažnijih svojstava niza ortogonalnih polinoma i predstavlja dragocenu informaciju pomoću koje možemo lako rekonstruisati ceo niz polinoma. Navodimo samo nekoliko najvažnijih svojstava tročlane rekurentne relacije.

- Mogućnost računanja nula polinoma π_n kao sopstvenih vrednosti prirdužene trodijagonalne matrice (poznate kao Jacobijeva matrica). Ove nule su veoma važne u konstrukciji Gaussovih kvadratura.
- Direktno izračunavanje normi polinoma $\|\pi_n\|$ koje su potrebne da bi se prešlo sa moničnih na ortonormirane polinome.
- Uspostavljanje veze izmedju ortogonalnih polinoma i verižnih razlomaka.

Poslednja primena koju smo spomenuli predstavlja osnovu metoda za računanje Hankelovih determinanti baziranog na verižnim razlomcima odnosno ortogonalnim polinomima.

Teorema 2.4.4. *Neka je $\{\pi_n\}_{n\in\mathbb{N}_0}$ niz moničnih ortogonalnih polinoma u odnosu na meru $d\lambda$. Tada ovaj niz zadovoljava sledeću tročlanu rekurentnu relaciju*

$$\pi_{n+1}(x) = (x - \alpha_n)\pi_n(x) - \beta_n\pi_{n-1}(x), \qquad (n \in \mathbb{N}_0), \tag{2.7}$$

sa startnim vrednostima $\pi_{-1}(x) = 0$ i $\pi_0(x) = 1$. Koeficijenti α_n i β_n mogu da se odrede iz sledećih relacija

$$\alpha_n = \frac{(x\pi_n, \pi_n)}{(\pi_n, \pi_n)}, \qquad (n \in \mathbb{N}_0),$$

$$\beta_n = \frac{(\pi_n, \pi_n)}{(\pi_{n-1}, \pi_{n-1})}, \qquad (n \in \mathbb{N}).$$

Pošto je $\pi_{-1}(x) = 0$ sledi da za koeficijent β_0 možemo izabrati proizvoljnu vrednost i da će relacija (2.7) ostati da važi. Najčešće se za ovaj koeficijent bira vrednost $\beta_0 = \|\pi_0\|^2$. Za ovako izabranu vrednost koeficijenta β_0 važi sledeća posledica

Posledica 2.4.5. *Neka su $\beta_0, \beta_1, \dots, \beta_n$ koeficijenti tročlane rekurentne relacije* (2.7)*. Tada važi*

$$\|\pi_n\|^2 = \beta_0\beta_1\cdots\beta_{n-1}\beta_n \qquad (n \in \mathbb{N}_0).$$

Napomenimo da se tročlana rekurentna relacija može iskoristiti za karakterizaciju niza moničnih ortogonalnih polinoma. Drugim rečima, ukoliko niz polinoma zadovoljava ovu relaciju sa zadatim startnim vrednostima, onda je taj niz polinoma ortogonalan u odnosu na neku meru $d\lambda$. Ovaj, na prvi pogled neočekivani rezultat je poznat u literaturi kao Favardova teorema (mada je poznat matematičkoj javnosti još od vremena kada je živeo i radio Stieltjes).

Teorema 2.4.6. (Favard) *Neka je $\{p_n(x)\}_{n\in\mathbb{N}_0}$ niz moničnih ortogonalnih polinoma koji zadovoljavaju $\deg p_n(x) = n$. Ovaj niz polinoma predstavlja niz ortogonalnih polinoma u odnosu na neku meru $d\lambda$ ako i samo ako postoje nizovi koeficijenata $\{\alpha_n\}_{n\in\mathbb{N}_0}$ i $\{\beta_n\}_{n\in\mathbb{N}_0}$ takvi da je zadovoljena tročlana rekurentna relacija* (2.7) *(uz pretpostavku da je $p_{-1}(x) = 0$).*

2.5 Metod za računanje Hankelove transformacije baziran na ortogonalnim polinomima

U ovom poglavlju ćemo detaljno opisati metod za izračunavanje Hankelove transformacije baziran na verižnim razlomcima, odnosno na ortogonalnim polinomima. Na kraju ćemo formulisati algoritam u kome ćemo taksativno navesti sve neophodne korake koji se moraju izvršiti da bi se došlo do izraza u zatvorenom obliku za Hankelovu transformaciju. U nastavku dajemo detaljniju analizu pojedinih koraka.

2.5.1 Veza Hankelove transformacije sa verižnim razlomcima i ortogonalnim polinomima

Sledeći rezultat, koji se pripisuje Heilermannu, uspostavlja vezu izmedju Hankelovih determinanti sa jedne i verižnih razlomaka sa druge strane. Sledeća teorema se može pronaći u [21, 146] a takodje je spomenuta od strane Krattenthalera u radu [71].

Teorema 2.5.1. [21, 71, 146] (Heilermann 1845) *Neka je $\{\mu_n\}_{n\in\mathbb{N}_0}$ niz brojeva čija je funkcija generatrisa jednaka $\mathcal{G}(x) = \sum_{n=0}^{+\infty} \mu_n x^n$ i može da se napiše u obliku*

$$\mathcal{G}(x) = \sum_{n=0}^{\infty} \mu_n x^n = \cfrac{\mu_0}{1 - \alpha_0 x - \cfrac{\beta_1 x^2}{1 - \alpha_1 x - \cfrac{\beta_2 x^2}{1 - \alpha_2 x - \cdots}}}. \tag{2.8}$$

Tada je Hankelova transformacija ovog niza $h = \mathbf{H}(\mu)$ odredjena sledećim izrazom

$$h_n = \mu_0^n \beta_1^{n-1} \beta_2^{n-2} \cdots \beta_{n-2}^2 \beta_{n-1}. \tag{2.9}$$

Teorema 2.5.1 omogućava eksplicitno izračunavanje Hankelove transformacije svakog niza $\{\mu_n\}_{n\in\mathbb{N}_0}$ čija funkcija generatrisa $G(x)$ može da se predstavi u obliku (2.8).

Primer 2.5.1. Primer ovakvog izračunavanja Hankelove transformacije, dat u radu [71], je Hankelova transformacija niza pomerenih Bernoullijevih brojeva $B'' = \{B_{n+2}\}_{n\in\mathbb{N}_0}$. Bernoullijevi brojevi su definisani eksponencijalnom funkcijom generatrise

$$\sum_{n=0}^{+\infty} \frac{B_n}{n!} z^n = \frac{z}{e^z - 1}.$$

Funkcija generatrisa pomerenih Bernoullijevih brojeva može se izraziti pomoću verižnog razlomka na sledeći način

$$\mathcal{G}(x) = \sum_{n=0}^{\infty} \mu_n x^n = \cfrac{1/6}{1 - \cfrac{\beta_1 x^2}{1 - \cfrac{\beta_2 x^2}{1 - \cdots}}}, \quad \beta_i = \frac{i(i+1)^2(i+2)}{4(2i+1)(2i+3)}, \quad (i \in \mathbb{N}).$$

Iz poslednjeg razvoja, korišćenjem relacije (2.9) možemo dobiti sledeći izraz u zatvorenom obliku za $h = \mathbf{H}(B'')$

$$h_n = (-1)^{\binom{n}{2}} \frac{1}{6} \prod_{i=1}^{n-1} \frac{i!(i+1)!^4(i+2)!}{(2i+2)!(2i+3)!}.$$

Još jedan primer izračunavanja Hankelove transformacije razvijanjem funkcije generatrise u verižni razlomak (2.8) dato je od strane Brualdija i Kirklanda u radu [12]. U tom radu, autori su izračunali u zatvorenom obliku izraz za Hankelovu transformaciju niza velikih Schröderovih brojeva. Takodje, još neki primeri ovakvog izračunavanja dati su u [71, 72].

Najčešće je veoma teško naći eksplicitan razvoj funkcije generatrise u verižni razlomak. Čak i da se pronadju (tj. naslute) izrazi za koeficijente u razvoju, veoma je teško dokazati ispravnost tog razvoja. Jedan od načina kako rešiti ovaj problem je uspostaviti vezu izmedju verižnih razlomaka oblika (2.8) sa jedne i ortogonalnih polinoma sa druge strane. Ovu vezu daje sledeća teorema.

Teorema 2.5.2. [71, 145, 146] *Neka je $\{p_n(x)\}_{n\in\mathbb{N}_0}$ niz moničnih ortogonalnih polinoma u odnosu na neki linearni funkcional $\mathcal{L}$. Posmatrajmo tročlanu rekurentnu relaciju*

$$p_{n+1}(x) = (x - \alpha_n)p_n(x) - \beta_n p_{n-1}(x).$$

Tada funkcija generatrisa momenata $\mathcal{G}(x) = \sum_{n=0}^{+\infty} \mu_n x^n$, gde su $\mu_n = \mathcal{L}(x^n)$ momenti zadovoljava (2.8) *pri čemu su α_n i β_n iz* (2.8) *upravo koeficijenti tročlane rekurentne relacije.*

Podsetimo se da smo koeficijent β_0 u tročlanoj rekurentnoj relaciji izabrali tako da je $\beta_0 = \mathcal{L}(1) = \mu_0$. Prema tome, relacija (2.9) može biti zapisana u obliku

$$h_n = \beta_0^n \beta_1^{n-1} \cdots \beta_{n-2}^2 \beta_{n-1}.$$

Sada ćemo u najkraćim crtama opisati kompletan metod za računanje Hankelove transformacije datog niza $\{\mu_n\}_{n\in\mathbb{N}_0}$. Na ovaj način je nastao Algoritam 2.5.1.

Algoritam 2.5.1 Metod za izračunavanje Hankelove transformacije baziran na ortogonalnim polinomima

Input: Niz $\{\mu_n\}_{n\in\mathbb{N}_0}$.

1: Naći meru $d\lambda$ (težinsku funkciju $w(x)$) takvu da je μ_n njen n-ti moment.
2: Naći koeficijente α_n i β_n tročlane rekurentne relacije koji odgovaraju meri $d\lambda$ (težini $w(x)$).
3: **return** $h_n := \beta_0^n \beta_1^{n-1} \cdots \beta_{n-2}^2 \beta_{n-1}$.

U naredna dva odeljka objasnićemo korake 1 i 2 Algoritma 2.5.1 detaljnije.

2.5.2 Izvodjenje izraza za težinsku funkciju korišćenjem Stieltjesove inverzione formule

U ovom odeljku razmatramo sledeći problem: Za dati niz $\{\mu_n\}_{n\in\mathbb{N}_0}$, potrebno je utvrditi da li postoji linearni funkcional $\mathcal{L}$ takav da je $\mathcal{L}(x^n) = \mu_n$ za svako $n \in \mathbb{N}_0$. Prema Rieszovoj reprezentacionoj teoremi, postojanje funkcionala $\mathcal{L}$ je ekvivalentno postojanju mere $d\lambda$ tako da važi $\int_{\mathbb{R}} x^n d\lambda = \mu_n$. Ovaj problem je u literaturi poznat kao *Hamburgerov momentni problem* [18]. Rešenje ovog problema dato je sledećom teoremom.

Teorema 2.5.3. *Za dati niz $\{\mu_n\}_{n\in\mathbb{N}_0}$, postoji pozitivna mera $d\lambda$ takva da važi $\int_{\mathbb{R}} x^n d\lambda = \mu_n$ za svako $n \in \mathbb{N}_0$, ako i samo ako je $h_n \geq 0$ za svako $n \in \mathbb{N}_0$ gde je $h = \{h_n\}_{n\in\mathbb{N}_0} = \mathbf{H}(\mu)$ Hankelova transformacija niza $\mu = \{\mu_n\}_{n\in\mathbb{N}_0}$.*

Prema tome, u ostatku ove glave pretpostavljamo da svi nizovi koje budemo razmatrali (i gde budemo koristili metod baziran na ortogonalnim polinomima) imaju nenegativnu Hankelovu transformaciju.

Postoje i drugi tipovi momentnih problema, zavisno od intervala integracije. Ako je interval $(0, +\infty)$, onda se radi o *Stieltjesovom momentnom problemu* a ako je interval $(0, 1)$ onda je to *Hausdorffov momentni problem*. Više o momentnim problemima može se naći na primer u [1, 11].

Sada ćemo razmotriti način za eksplicitno nalaženje rešenja Hamburgerovog momentnog problema. Za svaku meru $d\lambda$, definišimo njenu *Stieltjesovu transformaciju* na sledeći način

$$S(z; d\lambda) = \int_{\mathbb{R}} \frac{d\lambda(t)}{z - t}.$$

Neka je $\{\mu_n\}_{n\in\mathbb{N}_0}$ niz momenata mere $d\lambda$ i $G(z)$ funkcija generatrisa ovog niza. Tada važi

$$S(z; d\lambda) = \int_{\mathbb{R}} d\lambda z^{-1} \sum_{n=0}^{+\infty} z^{-n} t^n = z^{-1} \sum_{n=0}^{+\infty} z^{-n} \mu_n = z^{-1} G(z^{-1}).$$

Poslednja jednačina opisuje vezu izmedju Stieltjesove transformacije i funkcije generatrise momenata. Sada ćemo formulisati teormu, poznatu kao Stieltjes-Perronova inverziona formula (ili Stieltjesova inverziona formula) [17, 76] koja nam daje eksplicitno rešenje za meru $d\lambda$ (tj. funkciju $\lambda(t)$).

Teorema 2.5.4. [17, 76] (Stieltjes-Perronova inverziona formula) *Neka je $\{\mu_n\}_{n\in\mathbb{N}_0}$ niz takav da su svi elementi njegove Hankelove transformacije nenegativni. Označimo sa $G(z) = \sum_{n=0}^{n} \mu_n z^n$ funkciju generatrisu ovog niza i neka je $F(z) = z^{-1}G(z^{-1})$. Takodje, neka je funkcija $\lambda(t)$ definisana sledećim izrazom*

$$\lambda(t) - \lambda(0) = -\frac{1}{2\pi i} \lim_{y\to 0^+} \int_0^t \Big[F(x + iy) - F(x - iy) \Big] dx. \tag{2.10}$$

Tada važi $\mu_n = \int_{\mathbb{R}} x^n d\lambda$, tj. $\lambda(t)$ definiše rešenje Hamburgerovog momentnog problema.

Sledeća posledica će biti od koristi u narednim razmatranjima.

Posledica 2.5.5. *Neka su ispunjene pretpostavke predhodne teoreme i neka dodatno važi* $F(\bar{z}) = \overline{F(z)}$. *Tada je*

$$\lambda(t) - \lambda(0) = -\frac{1}{\pi} \lim_{y \to 0^+} \int_0^t \operatorname{Im} F(x+iy)dx. \tag{2.11}$$

2.5.3 Transformacije težinske funkcije

Već smo videli da je za izračunavanje Hankelove transformacije potrebno znati koeficijente $\tilde{\alpha}_n$ i $\tilde{\beta}_n$ tročlane rekurentne relacije koju zadovoljavaju monični polinomi ortogonalni u odnosu na neku težinu $\tilde{w}(x)$. Postoji više metoda za numeričko računanje ovih koeficijenata (videti npr. [39, 40]). U našem slučaju, potrebno je izračunati ove koeficijente u zatvorenom obliku. Jedan od načina za to je da se krene od neke druge težine $w(x)$ za koju znamo potrebne koeficijente α_n i β_n i da primenimo niz transformacija težinske funkcije tako da dobijemo traženu težinu $\tilde{w}(x)$. Prilikom izvodjenja opisanih transformacija potrebno je da znamo relacije koje povezuju originalne koeficijente α_n i β_n sa transformisanim $\tilde{\alpha}_n$ i $\tilde{\beta}_n$.

Na početku ćemo razmotriti dve veoma jednostavne transformacije.

Teorema 2.5.6. *Označimo sa* $w(x)$ *originalnu, sa* $\tilde{w}(x)$ *transformisanu težinsku funkciju a sa* $\{\pi_n(x)\}_{n\in\mathbb{N}_0}$ *i* $\{\tilde{\pi}_n(x)\}_{n\in\mathbb{N}_0}$ *nizove moničnih ortogonalnih polinoma u odnosu na originalnu i transformisanu težinsku funkciju respektivno. Takodje, označimo sa* $\{\alpha_n\}_{n\in\mathbb{N}_0}, \{\beta_n\}_{n\in\mathbb{N}_0}$ *i* $\{\tilde{\alpha}_n\}_{n\in\mathbb{N}_0}, \left\{\tilde{\beta}_n\right\}_{n\in\mathbb{N}_0}$ *odgovarajuće koeficijente tročlane rekurentne relacije za originalnu i transformisanu težinsku funkciju respektivno.*

Važe sledeće transformacione formule:

(1) *Ako je* $\tilde{w}(x) = Cw(x)$ *gde je* $C > 0$ *onda važi* $\tilde{\alpha}_n = \alpha_n$ *za* $n \in \mathbb{N}_0$ *i* $\tilde{\beta}_0 = C\beta_0$, $\tilde{\beta}_n = \beta_n$ *za* $n \in \mathbb{N}$. *Dodatno važi* $\tilde{\pi}_n(x) = \pi_n(x)$ *za svako* $n \in \mathbb{N}_0$.

(2) *Ako je* $\tilde{w}(x) = w(ax+b)$ *gde je* $a, b \in \mathbb{R}$ *i* $a \neq 0$ *onda važi* $\tilde{\alpha}_n = \frac{\alpha_n - b}{a}$ *za* $n \in \mathbb{N}_0$ *i* $\tilde{\beta}_0 = \frac{\beta_0}{|a|}$ *i* $\tilde{\beta}_n = \frac{\beta_n}{a^2}$ *za* $n \in \mathbb{N}$. *Pored toga važi* $\tilde{\pi}_n(x) = \frac{1}{a^n}\pi_n(ax+b)$.

Dokaz.
(1) Sledeća relacija je zadovoljena

$$\int_R \tilde{\pi}_n(x)\tilde{\pi}_k(x)\tilde{w}(x)dx = C\int_R \pi_n(x)\pi_k(x)w(x)dx = 0$$

za svako $n, k \in \mathbb{N}_0$ za koje je $n \neq k$. Ovim smo dokazali ortogonalnost polinoma $\{\pi_n\}_{n\in\mathbb{N}_0}$. Prema tome važi $\tilde{\alpha}_n = \alpha_n$ za $n \in \mathbb{N}_0$ i $\tilde{\beta}_n = \beta_n$ za $n \in \mathbb{N}$. Koeficijent $\tilde{\beta}_0$ je jednak

$$\tilde{\beta}_0 = \int_{\mathbb{R}} \tilde{w}(x)dx = C\int_{\mathbb{R}} w(x)dx = C\beta_0.$$

Ovim je završen dokaz dela **(1)** teoreme.

(2) Neka je $\tilde{\pi}_n(x) = \frac{1}{a^n}\pi_n(ax+b)$. Onda je

$$\begin{aligned}\int_{\mathbb{R}} \tilde{\pi}_n(x)\tilde{\pi}_k(x)\tilde{w}(x)dx &= \frac{1}{a^{n+k+1}}\int_{\mathbb{R}} \pi_n(ax+b)\pi_k(ax+b)w(ax+b)dx \\ &= \frac{1}{a^{n+k+1}}\int_{\mathbb{R}} \pi_n(y)\pi_k(y)w(y)dy = 0\end{aligned}$$

za svako $n, k \in \mathbb{N}_0$ i $n \neq k$. Pored toga važi

$$\begin{aligned}\tilde{\pi}_{n+1}(x) &= \frac{1}{a^{n+1}}\pi_{n+1}(ax+b) = \frac{1}{a^{n+1}}[(ax+b-\alpha_k)\pi_n(ax+b) - \beta_n\pi_{n-1}(ax+b)] \\ &= \left(x - \frac{\alpha_n - b}{a}\right)\tilde{\pi}_n(x) - \frac{\beta_n}{a^2}\tilde{\pi}_{n-1}(x)\end{aligned}$$

odakle možemo zaključiti da je $\tilde{\alpha}_n = \frac{\alpha_n - b}{a}$ za $n \in \mathbb{N}_0$ i $\tilde{\beta}_n = \frac{\beta_n}{a^2}$ za $n \in \mathbb{N}$. Ponovo, direktnim izračunavanjem dobijamo

$$\tilde{\beta}_0 = \int_{\mathbb{R}} \tilde{w}(x)dx = \int_{\mathbb{R}} w(ax+b)dx = \frac{\beta_0}{|a|}.$$

Ovim je završen dokaz dela **(2)** teoreme. □

Sada ćemo razmotriti transformacije oblika

$$\tilde{w}(x) = \frac{u(x)}{v(x)}w(x), \qquad u(x), v(x) \in \mathbb{R}[x].$$

Naredni rezultati su uglavnom preuzeti iz knjige [39]. Osnova tih rezultata je sledeća generalizacija poznate Christoffelove teoreme.

Teorema 2.5.7. *Neka su $\pi_n(x)$ i $\tilde{\pi}_n(x)$, $n \in \mathbb{N}_0$ monični ortogonalni polinomi u odnosu na težine $w(x)$ i $\tilde{w}(x) = r(x)w(x)$ respektivno, gde je $r(x) = u(x)/v(x)$ i $u(x) = \prod_{i=1}^{l}(x-u_i)$, $v(x) = \prod_{j=1}^{m}(x-v_j)$. U slučaju $m \leq n$, važi*

$$u(x)\tilde{\pi}_n(x) = C\det\begin{bmatrix} \pi_{n-m}(x) & \cdots & \pi_{n-1}(x) & \pi_n(x) & \cdots & \pi_{n+l}(x) \\ \pi_{n-m}(u_1) & \cdots & \pi_{n-1}(u_1) & \pi_n(u_1) & \cdots & \pi_{n+l}(u_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \pi_{n-m}(u_l) & \cdots & \pi_{n-1}(u_l) & \pi_n(u_l) & \cdots & \pi_{n+l}(u_l) \\ \rho_{n-m}(v_1) & \cdots & \rho_{n-1}(v_1) & \rho_n(v_1) & \cdots & \rho_{n+l}(v_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_{n-m}(v_m) & \cdots & \rho_{n-1}(v_m) & \rho_n(v_m) & \cdots & \rho_{n+l}(v_m) \end{bmatrix}. \tag{2.12}$$

U suprotnom, ako je $m > n$ onda važi

$$u(x)\tilde{\pi}_n(x) = C\det\begin{bmatrix} 0 & 0 & \cdots & 0 & \pi_0(t) & \cdots & \pi_{n+l}(t) \\ 0 & 0 & \cdots & 0 & \pi_0(u_1) & \cdots & \pi_{n+l}(u_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & \pi_0(u_l) & \cdots & \pi_{n+l}(u_l) \\ 1 & v_1 & \cdots & v_1^{m-n-1} & \rho_0(v_1) & \cdots & \rho_{n+l}(v_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & v_m & \cdots & v_m^{m-n-1} & \rho_0(v_m) & \cdots & \rho_{n+l}(v_m) \end{bmatrix}. \tag{2.13}$$

gde je C normalizaciona konstanta (jednaka je recipročnoj vrednosti vodećeg koeficijenta polinoma sa desne strane). Sa $\rho_n(z)$ označili smo Cauchyjeve integrale polinoma $\pi_n(x)$ definisane sa

$$\rho_n(z) = \int_{\mathbb{R}} \frac{\pi_n(t)}{z-t} w(t)dt, \qquad (n \in \mathbb{N}_0).$$

Originalna Cristoffelova teorema se dobija u slučaju $v(x) = 1$, odnosno $m = 0$. Transformacione formule dobićemo primenom Teoreme 2.5.7 u specijalnim slučajevima.

Teorema 2.5.8. *Posmatrajmo istu notaciju kao u Teoremi 2.5.6. Neka je niz $\{r_n\}_{n\in\mathbb{N}_0}$ definisan sa*

$$r_0 = c - \alpha_0, \qquad r_n = c - \alpha_n - \frac{\beta_n}{r_{n-1}} \qquad (n \in \mathbb{N}). \tag{2.14}$$

Razmotrimo dva slučaja:

(1) *Ako je $\tilde{w}(x) = (x-c)w(x)$ gde je $c < \inf \operatorname{supp}(w)$, tada važi*

$$\begin{aligned} &\tilde{\beta}_0 = \int_{\mathbb{R}} \tilde{w}(x)dx, && \tilde{\beta}_n = \beta_n \frac{r_n}{r_{n-1}}, \qquad (n \in \mathbb{N}), \\ &\tilde{\alpha}_n = \alpha_{n+1} + r_{n+1} - r_n, \qquad (n \in \mathbb{N}_0) \end{aligned}$$

(2) *Ako je $\tilde{w}(x) = (x-c)(x-\bar{c})w(x)$ gde je $c \in \mathbb{C} \setminus \mathbb{R}$, tada važi*

$$\begin{aligned} &\tilde{\beta}_0 = \beta_0(\beta_1 + |r_0|^2), \qquad \tilde{\beta}_n = \beta_n \frac{r''_{n+1} r''_{n-1}}{(r''_{n-1})^2} \left| \frac{r_n}{r_{n-1}} \right|^2 \qquad (n \in \mathbb{N}), \\ &\tilde{\alpha}_n = \alpha_{n+2} + r'_{n+2} + \frac{r''_{n+2}}{r''_{n+1}} r'_{n+1} - \left(r'_{n+1} + \frac{r''_{n+1}}{r''_n} r'_n \right) \qquad (n \in \mathbb{N}_0). \end{aligned}$$

gde je $r'_n = \operatorname{Re} r_n$ i $r''_n = \operatorname{Im} r_n$.

Dokaz. Pretpostavimo da je $\tilde{w}(x) = (x-c)w(x)$ gde je $c < \inf \operatorname{supp}(w)$. Primenom Teoreme 2.5.7 dobijamo

$$(x-c)\tilde{\pi}_n(x) = -\frac{1}{\pi_n(c)} \det \begin{bmatrix} \pi_n(x) & \pi_{n+1}(x) \\ \pi_n(c) & \pi_{n+1}(c) \end{bmatrix} = \pi_{n+1}(x) - r_n \pi_n(t),$$

gde smo sa r_n označili $r_n = \pi_{n+1}(c)/\pi_n(c)$. Napisaćemo polinom $(x-c)x\tilde{\pi}_n(x)$ na dva načina. Najpre ćemo koristiti tročlanu rekurentnu relaciju za niz $\{\pi_n(x)\}_{n\in\mathbb{N}_0}$. Imamo da je

$$\begin{aligned} (x-c)x\tilde{\pi}_n(x) &= x\pi_{n+1}(x) - r_n x \pi_n(x) \\ &= \pi_{n+2}(x) + (\alpha_{n+1} - r_n)\pi_{n+1}(x) + (\beta_{n+1} - r_n\alpha_n)\pi_n(x) - r_n\beta_n\pi_{n-1}(x). \end{aligned} \tag{2.15}$$

Zatim ćemo iskoristiti tročlanu rekurentnu relaciju za niz $\{\tilde{\pi}_n(x)\}_{n\in\mathbb{N}_0}$. Na taj način dobijamo

$$\begin{aligned} (x-c)x\tilde{\pi}_n(x) &= (x-c)[\tilde{\pi}_{n+1}(x) + \tilde{\alpha}_n\tilde{\pi}_n(x) + \tilde{\beta}_n\tilde{\pi}_{n-1}(x)] \\ &= \pi_{n+2}(x) + (\tilde{\alpha}_n - r_{n+1})\pi_{n+1}(x) + (\tilde{\beta}_n - r_n\tilde{\alpha}_n)\pi_n(x) - r_{n-1}\tilde{\beta}_n\pi_{n-1}(x). \end{aligned} \tag{2.16}$$

Poredjenjem koeficijenta u (2.15) kao i u (2.16) dobijamo sledeće relacije

$$\tilde{\alpha}_n - r_{n+1} = \alpha_{n+1} - r_n, \qquad r_{n-1}\tilde{\beta}_n = r_n\beta_n. \tag{2.17}$$

Ovim smo dokazali deo **(1)** teoreme. Deo **(2)** se dokazuje analogno. □

U nekim slučajevima je veoma teško naći rešenje u zatvorenom obliku diferencne jednačine (2.14). Sledeća posledica uprošćava transformacione formule iz dela **(1)** poslednje teoreme uvodeći nov pomoćni niz.

Posledica 2.5.9. *Posmatrajmo istu notaciju kao u Teoremi 2.5.6. Neka je niz* $\{\lambda_n\}_{n\in\mathbb{N}_0}$ *definisan sa*

$$\lambda_{-1} = 0, \quad \lambda_0 = 1, \quad \lambda_{n+1} = (c-\alpha_n)\lambda_n - \beta_n\lambda_{n-1} \qquad (n \in \mathbb{N}_0). \tag{2.18}$$

gde je $c < \inf \operatorname{supp}(w)$*. Tada ako važi* $\tilde{w}(x) = (x-c)w(x)$ *imamo da je*

$$\begin{aligned} \tilde{\beta}_0 &= \int_{\mathbb{R}} \tilde{w}(x)dx, & \tilde{\beta}_n &= \beta_n \frac{\lambda_{n+1}\lambda_{n-1}}{\lambda_n^2} \qquad (n \in \mathbb{N}), \\ \tilde{\alpha}_n &= c - \frac{\lambda_{n+1}}{\lambda_n} - \beta_{n+1}\frac{\lambda_n}{\lambda_{n+1}} \qquad (n \in \mathbb{N}_0). \end{aligned} \tag{2.19}$$

Dokaz. Neka je $\lambda_n = \pi_n(c)$. Kako važi $r_n = \lambda_{n+1}/\lambda_n$ i niz $\{\lambda_n\}_{n\in\mathbb{N}_0}$ zadovoljava rekurentnu relaciju

$$\lambda_{n+1} = (c-\alpha_n)\lambda_n - \beta_n\lambda_{n-1},$$

dobijamo traženi rezultat. □

Pretpostavimo da je transformacija zadata pomoću $\tilde{w}(x) = \frac{w(x)}{x-c}$ gde je $c < \inf \operatorname{supp}(w)$. Na osnovu Teoreme 2.5.7 važi

$$\tilde{\pi}_n(x) = -\frac{1}{\rho_{n-1}(c)} \det \begin{bmatrix} \pi_{n-1}(x) & \pi_n(x) \\ \rho_{n-1}(c) & \rho_n(c) \end{bmatrix} = \pi_n(x) - r_{n-1}\pi_{n-1}(x)$$

gde je sad $r_n = \rho_{n+1}(c)/\rho_n(c)$. Može se dokazati da Cauchyjevi integrali $\rho_n(z)$ zadovoljavaju istu rekurentnu relaciju kao i polinomi $\pi_n(x)$. Razlika je samo u početnim uslovima, pošto važi $\rho_{-1}(x) = 1$ i $\rho_0(x) = \int_{\mathbb{R}} \frac{w(t)}{t-x}dt$. Sada ćemo dokazati ovu činjenicu. Imamo da je

$$\begin{aligned} \rho_{n+1}(x) &= \int_{\mathbb{R}} \frac{(x-\alpha_n)\pi_n(t) - \beta_n\pi_{n-1}(t)}{x-t} w(t)dt = \int_{\mathbb{R}} \frac{(t-\alpha_n)\pi_n(t)}{x-t} w(t)dt - \beta_n\rho_{n-1}(x) \\ &= -\int_{\mathbb{R}} \pi_n(t)w(t)dt + (x-\alpha_n)\rho_n(x) - \beta_n\rho_{n-1}(x) \\ &= (x-\alpha_n)\rho_n(x) - \beta_n\rho_{n-1}(x), \qquad (n \in \mathbb{N}). \end{aligned}$$

Za $n = 0$ takodje važi

$$\rho_1(x) = (x-\alpha_0)\rho_0(x) - \beta_0\rho_{-1}(x),$$

pošto je $\rho_{-1}(x) = 1$. Prema tome niz $\{r_n\}_{n\in\mathbb{N}_0}$ zadovoljava istu rekurentnu relaciju kao u prvom slučaju, samo što se razlikuju startne vrednosti

$$r_n = c - \alpha_n - \frac{\beta_n}{r_{n-1}} \quad (n \in \mathbb{N}), \qquad r_{-1} = -\int_{\mathbb{R}} \tilde{w}(t)dt.$$

Ponavljanjem iste procedure kao u predhodnom slučaju (sada predstavljamo na dva načina polinom $(t-x)\pi_n(t)$) dokazujemo deo **(1)** sledeće teoreme. Drugi deo se dokazuje analogno.

Teorema 2.5.10. *Posmatrajmo istu notaciju kao u Teoremi 2.5.6. Neka je niz* $\{r_n\}_{n\in\mathbb{N}_0}$ *definisan sa*

$$r_{-1} = -\int_{\mathbb{R}} \tilde{w}(x)\ dx, \qquad r_n = c - \alpha_n - \frac{\beta_n}{r_{n-1}} \qquad (n \in \mathbb{N}_0). \tag{2.20}$$

Razlikovaćemo dva slučaja.

(1) *Ako je* $\tilde{w}(x) = \frac{w(x)}{x-c}$ *gde je* $c < \inf \operatorname{supp}(w)$ *tada važi*

$$\begin{aligned} &\tilde{\alpha}_0 = \alpha_0 + r_0, && \tilde{\alpha}_n = \alpha_n + r_n - r_{n-1}, \\ &\tilde{\beta}_0 = -r_{-1}, && \tilde{\beta}_n = \beta_{n-1}\frac{r_{n-1}}{r_{n-2}}, \qquad (n \in \mathbb{N}) \end{aligned} \tag{2.21}$$

(2) *Ako je* $\tilde{w}(x) = \frac{w(x)}{c-x}$ *gde je* $c > \sup \operatorname{supp}(w)$, *tada važe važe relacije* (2.20) *i* (2.21) *pri čemu je sad* $\tilde{\beta}_0 = r_{-1}$ *gde je*

$$r_{-1} = \int_{\mathbb{R}} \tilde{w}(x)\ dx.$$

2.6 Hankelova transformacija sume dva uzastopna generalisana Catalanova broja

U ovom odeljku razmotrićemo Hankelovu transformaciju niza čiji su elementi sume dva uzastopna generalisana Catalanova broja. Glavni rezultat ovog poglavlja je izraz u zatvorenom obliku koji predstavlja opšti član Hankelove transformacije posmatranog niza. Rezultati prikazani u ovom poglavlju su originalni i preuzeti iz našeg rada [113].

2.6.1 Uvod

Razmotrićemo niz čiji je opšti član suma dva uzastopna generalisana Catalanova broja koji zavise od parametra L

$$a_0(L) = L + 1, \qquad a_n(L) = c(n; L) + c(n+1; L) \qquad (n \in \mathbb{N}), \tag{2.22}$$

gde je

$$c(n; L) = T(2n, n; L) - T(2n, n-1; L), \tag{2.23}$$

Sa $T(n,k;L)$ ovde smo označili sledeću generalizaciju binomnih koeficijenata

$$T(n,k;L) = \sum_{j=0}^{n-k} \binom{k}{j}\binom{n-k}{j} L^j. \tag{2.24}$$

Napomenimo da se niz brojeva $\{T(n,k;L)\}_{n,k\in\mathbb{N}_0}$ naziva *generalisani Pascalov trougao.* Najpre dajemo nekoliko osnovnih svojstava niza $\{a_n(L)\}_{n\in\mathbb{N}_0}$ a zatim formulišemo glavni rezultat kojim se odredjuje Hankelova transformacija ovog niza. U sledećem primeru razmatramo specijalan slučaj kada je $L = 1$.

Primer 2.6.1. Neka je $L = 1$. Tada se generalisani Catalanovi brojevi svode na obične Catalanove brojeve. Ovo važi na osnovu Vandermondeovog konvolucionog identiteta

$$\binom{n}{k} = \sum_j \binom{k}{j}\binom{n-k}{j}.$$

Korišćenjem sledeće jednakosti

$$T(2n,n;1) = \binom{2n}{n}, \qquad T(2n,n-1;1) = \binom{2n}{n-1},$$

dobija se

$$C(n) = \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}$$

kao i

$$a_n(1) = C(n) + C(n+1) = \frac{(2n)!(5n+4)}{n!(n+2)!} \qquad (n = 0,1,2,\ldots).$$

Hankelova transformacija niza $\{a_n(1)\}_{n\in\mathbb{N}_0}$ je već izračunata u zatvorenom obliku u radu Cvetkovića, Rajkovića i Ivkovića [26]. Autori ovog rada su dokazali da je Hankelova transformacija niza $\{a_n(1)\}_{n\in\mathbb{N}_0}$ jednaka nizu Fibonaccijevih brojeva sa neparnim indeksima

$$h_n(1) = F_{2n+1} = \frac{1}{\sqrt{5}\ 2^{n+1}}\left\{(\sqrt{5}+1)(3+\sqrt{5})^n + (\sqrt{5}-1)(3-\sqrt{5})^n\right\}.$$

Motivacija za ovaj rad proistekla je iz hipoteze koju je postavio Paul Barry [7].

Primer 2.6.2. Za $L = 2$, prvih nekoliko članova niza $\{a_n(2)\}_{n\in\mathbb{N}_0}$ su sledeći brojevi

$$3, 8, 28, 112, 484, \ldots,$$

dok su prvih nekoliko članova njihove Hankelove transformacije $\{h_n\}_{n\in\mathbb{N}_0}$ jednaki

$$3, 20, 272, 7424, 405504, \ldots.$$

P. Barry je postavio hipotezu [7] da važi

$$h_n(2) = 2^{\frac{n^2-n}{2}-2}\left\{(2+\sqrt{2})^{n+1} + (2-\sqrt{2})^{n+1}\right\}.$$

Barryjeva hipoteza može se jednostavno uopštiti u sledeći glavni rezultat koji dokazujemo kroz celo ovo poglavlje.

Teorema 2.6.1. *Za generalisani Paskalov trougao* $\{T(n,k;L)\}_{n,k\in\mathbb{N}_0}$*, Hankelova transformacija niza čiji je opšti član jednak sumi dva uzastopna generalisana Catalanova broja,* $a_n(L) = c(n;L) + c(n+1;L)$ *dat je sledećim izrazom*

$$
\begin{aligned}
h_n(L) = {} & \frac{L^{(n^2-n)/2}}{2^{n+1}\sqrt{L^2+4}} \\
& \times \left\{(\sqrt{L^2+4}+L)(\sqrt{L^2+4}+L+2)^n + (\sqrt{L^2+4}-L)(L+2-\sqrt{L^2+4})^n\right\}.
\end{aligned}
\tag{2.25}
$$

Uvedimo sledeće oznake koje ćemo koristiti do kraja ovog poglavlja

$$\xi = \sqrt{L^2+4}, \quad t_1 = L+2+\xi, \quad t_2 = L+2-\xi. \tag{2.26}$$

Izraz (2.25) možemo zapisati u sledećem obliku

$$h_n(L) = \frac{L^{n(n-1)/2}}{2^{n+1}\xi}\left((\xi+L)t_1^n + (\xi-L)t_2^n\right).$$

Dalje, uvodjenjem sledeća dva pomoćna niza

$$\varphi_n = t_1^n + t_2^n, \quad \psi_n = t_1^n - t_2^n \qquad (n \in \mathbb{N}_0), \tag{2.27}$$

relaciju (2.25) možemo dalje uprostiti, tako da dobijamo

$$h_n(L) = \frac{L^{n(n-1)/2}}{2^{n+1}\xi}\cdot\left(L\psi_n + \xi\varphi_n\right). \tag{2.28}$$

Sledeća lema daje neka osnovna svojstva nizova φ_n i ψ_n koja će biti od koristi prilikom uprošćavanja izraza za $h_n(L)$ do konačnog oblika definisanog izrazom (2.25).

Lema 2.6.2. *Vrednosti* φ_n *i* ψ_n *zadovoljavaju sledeće izraze*

$$\varphi_j\cdot\varphi_k = \varphi_{j+k} + (4L)^j\varphi_{k-j}, \quad \psi_j\cdot\psi_k = \varphi_{j+k} - (4L)^j\varphi_{k-j} \qquad (0\le j\le k) \tag{2.29}$$

$$\varphi_j\cdot\psi_k = \psi_{j+k} + (4L)^j\psi_{k-j}, \qquad \psi_j\cdot\varphi_k = \psi_{j+k} - (4L)^j\psi_{k-j} \qquad (0\le j\le k). \tag{2.30}$$

Iako to nije očigledno iz relacije (2.25), konačan izraz za $h_n(L)$ je polinom po promenljivoj L. Sledeća posledica dokazuje ovu činjenicu i daje polinomnu reprezentaciju za $h_n(L)$.

Posledica 2.6.3. *Ukoliko je glavna teorema tačna, izraz za* $h_n(L)$ *može da se napiše u obliku sledećeg polinoma*

$$
\begin{aligned}
h_n(L) = {} & 2^{-n}L^{n(n-1)/2} \\
& \cdot\left\{\sum_{i=0}^{[(n-1)/2]}\binom{n}{2i+1}L(L+2)^{n-2i-1}(L^2+4)^i + \sum_{i=0}^{[n/2]}\binom{n}{2i}(L+2)^{n-2i}(L^2+4)^i\right\}.
\end{aligned}
$$

Dokaz. Korišćenjem predhodno uvedene notacije možemo pisati

$$
\begin{aligned}
&(L+\xi)(L+2+\xi)^n-(L-\xi)(L+2-\xi)^n\\
&=(L+\xi)\sum_{k=0}^{n}\tbinom{n}{k}(L+2)^{n-k}\xi^k-(L-\xi)\sum_{k=0}^{n}(-1)^k\tbinom{n}{k}(L+2)^{n-k}\xi^k\\
&=\sum_{k=0}^{n}\big(1-(-1)^k\big)\tbinom{n}{k}L(L+2)^{n-k}\xi^k+\sum_{k=0}^{n}\big(1+(-1)^k\big)\tbinom{n}{k}(L+2)^{n-k}\xi^{k+1}\\
&=2\sum_{i=0}^{\lfloor(n-1)/2\rfloor}\tbinom{n}{2i+1}L(L+2)^{n-2i-1}\xi^{2i+1}+2\sum_{i=0}^{\lfloor n/2\rfloor}\tbinom{n}{2i}(L+2)^{n-2i}\xi^{2i+1}\\
&=2\xi\left\{\sum_{i=0}^{\lfloor(n-1)/2\rfloor}\tbinom{n}{2i+1}L(L+2)^{n-2i-1}\xi^{2i}+\sum_{i=0}^{\lfloor n/2\rfloor}\tbinom{n}{2i}(L+2)^{n-2i}\xi^{2i}\right\},
\end{aligned}
$$

odakle direktno sledi pretpostavljena polinomna reprezentacija za niz $h_n(L)$. $\square$

2.6.2 Funkcija generatrisa

Izvešćemo izraz za funkciju generatrisu niza $\{h_n(L)\}_{n\in\mathbb{N}_0}$. Pritom će nam od koristi biti izraz za funkciju generatrisu Jacobijevih polinoma $P_n^{(a,b)}(x)$. Jacobijevi polinomi su definisani na sledeći način

$$
P_n^{(a,b)}(x)=\frac{1}{2^n}\sum_{k=0}^{n}\binom{n+a}{k}\binom{n+b}{n-k}(x-1)^{n-k}(x+1)^k\quad(a,b>-1).
$$

Takodje, možemo ih zapisati u sledećem obliku

$$
P_n^{(a,b)}(x)=\left(\frac{x-1}{2}\right)^n\sum_{k=0}^{n}\binom{n+a}{k}\binom{n+b}{n-k}\left(\tfrac{x+1}{x-1}\right)^k.
$$

Iz činjenice da važi

$$
L=\frac{x+1}{x-1}\quad\Leftrightarrow\quad x=\frac{L+1}{L-1}\qquad(x\neq1,\ L\neq1).
$$

možemo zaključiti da je

$$
\begin{aligned}
T(2n,n;L)&=(L-1)^n\cdot P_n^{(0,0)}\left(\tfrac{L+1}{L-1}\right),\\
T(2n+2,n;L)&=(L-1)^n\cdot P_n^{(2,0)}\left(\tfrac{L+1}{L-1}\right).
\end{aligned}
$$

Funkcija generatrisa $G(x,t)$ Jakobijevih polinoma može se zapisati na sledeći način

$$
G^{(a,b)}(x,t)=\sum_{n=0}^{\infty}P_n^{(a,b)}(x)t^n=\frac{2^{a+b}}{\phi\cdot(1-t+\phi)^a\cdot(1+t+\phi)^b}. \tag{2.31}
$$

Pritom smo uveli sledeću oznaku

$$
\phi=\phi(x,t)=\sqrt{1-2xt+t^2}.
$$

Sada važi

$$\sum_{n=0}^{\infty} T(2n,n;L)\ t^n = \sum_{n=0}^{\infty} P_n^{(0,0)}\Big(\tfrac{L+1}{L-1}\Big)\big((L-1)t\big)^n = G^{(0,0)}\Big(\tfrac{L+1}{L-1},(L-1)t\Big),$$
$$\sum_{n=0}^{\infty} T(2n+2,n;L)\ t^n = \sum_{n=0}^{\infty} P_n^{(2,0)}\Big(\tfrac{L+1}{L-1}\Big)\big((L-1)t\big)^n = G^{(2,0)}\Big(\tfrac{L+1}{L-1},(L-1)t\Big).$$

Takodje važi i

$$\sum_{n=0}^{\infty} T(2n,n-1;L)\ t^n = t\cdot\Big\{G^{(2,0)}\Big(\tfrac{L+1}{L-1},(L-1)t\Big)-1\Big\},$$
$$\sum_{n=0}^{\infty} T(2n+2,n+1;L)\ t^n = \frac{1}{t}\cdot\Big\{G^{(0,0)}\Big(\tfrac{L+1}{L-1},(L-1)t\Big)-1\Big\}.$$

Konačno, funkcija generatrisa $\mathcal{G}(t;L)$ niza $\{a_n(L)\}_{n\in\mathbb{N}_0}$ je data pomoću

$$\begin{aligned}\mathcal{G}(t;L) &= \sum_{n=0}^{\infty} a_n(L)t^n \\ &= \frac{t+1}{t}G^{(0,0)}\Big(\tfrac{L+1}{L-1},(L-1)t\Big)-(t+1)G^{(2,0)}\Big(\tfrac{L+1}{L-1},(L-1)t\Big)-\frac{1}{t}.\end{aligned} \tag{2.32}$$

Posle malo računanja i sredjivanja predhodnog izraza dobijamo da važi sledeća teorema

Teorema 2.6.4. *Funkcija generatrisa $\mathcal{G}(t;L)$ niza $\{a_n(L)\}_{n\in\mathbb{N}_0}$ data je pomoću*

$$\mathcal{G}(t;L) = \frac{t+1}{\rho(t;L)}\ \left\{\frac{1}{t}-\frac{4}{(1-(L-1)t+\rho(t;L))^2}\right\}-\frac{1}{t}, \tag{2.33}$$

gde je

$$\rho(t;L) = \phi\Big(\tfrac{L+1}{L-1},(L-1)t\Big) = \sqrt{1-2(L+1)t+(L-1)^2t^2} \tag{2.34}$$

Funkcija $\rho(t;L)$ ima domen

$$D_\rho = \Big(-\infty,\frac{1-2\sqrt{L}+L}{1-2L+L^2}\Big)\cup\Big(\frac{1+2\sqrt{L}+L}{1-2L+L^2},+\infty\Big)\qquad (L\neq 1),$$

kao i

$$D_\rho = (-\infty,1/4)\qquad (L=1).$$

Primer 2.6.3. Za $L=1$, dobijamo

$$\mathcal{G}(t;1) = \sum_{n=0}^{\infty} a_n(1)\ t^n = \frac{1}{t}\left(\frac{(1-\sqrt{1-4t})(1+t)}{2t}-1\right). \tag{2.35}$$

dok za $L=2$ važi

$$\mathcal{G}(t;2) = \sum_{n=0}^{\infty} a_n(2)\ t^n = -\frac{1}{t}+\frac{t+1}{\sqrt{t^2-6t+1}}\left\{\frac{1}{t}-\frac{4}{(1-t+\sqrt{t^2-6t+1})^2}\right\}. \tag{2.36}$$

2.6.3 Odredjivanje težinske funkcije

Korišćenjem Stieltjesove inverzione forumule (2.11) možemo odrediti izraz za težinsku funkciju $\omega(x;L)$ čiji su momenti elementi niza $\{a_n(L)\}_{n\in\mathbb{N}_0}$.

Teorema 2.6.5. *Težinska funkcija $\omega(x;L)$ čiji je niz momenata jednak $\{a_n(L)\}_{n\in\mathbb{N}_0}$ data je sledećim izrazom*

$$\omega(x;L) = \begin{cases} \frac{\sqrt{L}}{\pi}\left(1+\frac{1}{x}\right)\sqrt{1-\left(\frac{x-L-1}{2\sqrt{L}}\right)^2}, & x\in((\sqrt{L}-1)^2,(\sqrt{L}+1)^2) \\ 0, & \textit{inače} \end{cases} \tag{2.37}$$

Dokaz. Uvedimo funkciju $F(z;L)$ na osnovu

$$F(z;L) = \sum_{k=0}^{\infty} a_k(L) z^{-k-1} = z^{-1}\mathcal{G}(z^{-1};L).$$

Korišćenjem već izvedenog izraza (2.33) za funkciju generatrisu $\mathcal{G}(t;L)$ i posle sredjivanja dobijamo

$$\begin{aligned} F(z;L) &= -1 + \frac{2(z+1)}{L-1+z+\sqrt{L^2+(z-1)^2-2L(z+1)}} \\ &= -1 + \frac{2(z+1)}{L-1+z(1+z\rho(z^{-1},L))}. \end{aligned} \tag{2.38}$$

Direktnom proverom možemo da utvrdimo da važi $F(\overline{z};L) = \overline{F(z;L)}$. Posmatrajmo funkciju $F(z;L)$ u gornjoj poluravni $\{z\in\mathbb{C} \mid \mathrm{Im}\ z>0\}$ kompleksne ravni $\mathbb{C}$. Funkcija

$$R(z;L) = z\rho(z^{-1},L) = \sqrt{L^2+(z-1)^2-2L(z+1)}.$$

ima dve tačke grananja, $(\sqrt{L}-1)^2$ kao i $(\sqrt{L}+1)^2$. Izabraćemo regularnu granu korena u predhodnom izrazu tako da on ima pozitivnu vrednost kada je izraz pod korenom pozitivan.

Potrebno je naći integral funkcije $F(z;L)$. Eksplicitnim izračunavanjem (korišćenjem mogućnosti programskog paketa MATHEMATICA) dobijamo da važi

$$\mathcal{F}(z;L) = \int F(z;L)dz = \frac{1}{4}\Big[z^2-2Lz-(z-L+1)R(z;L)-l_1(z)+l_2(z)\Big], \tag{2.39}$$

gde je

$$l_1(z) = 2(3L+1)\log\Big[z-(L+1)+R(z;L)\Big], \tag{2.40}$$

$$l_2(z) = 2(L-1)\log\Big[\frac{-(L-1)R(z;L)-(L-1)^2+z(L+1)}{z^2(L-1)^3}\Big] \tag{2.41}$$

Funkcije $l_1(z;L)$ i $l_2(z;L)$ imaju još jednu tačku grananja, $z=L+1$. Izabraćemo regularnu granu logaritma u izrazima (2.40) i (2.41) tako da je imaginarni deo jednak 0 ukoliko je vrednost pod logaritmom pozitivna i realna.

Ako funkciju $R(z;L)$ napišemo u obliku

$$R(z;L) = \sqrt{(z-L-1)^2-4L}$$

i zamenimo $z = x+iy$, dobijamo

$$R(x;L) = \lim_{y\to 0^+} R(x+iy;L) = \begin{cases} i\sqrt{4L-(x-L-1)^2}, & x\in(a,b),\\ \sqrt{(x-L-1)^2-4L}, & \text{inače} \end{cases}$$

gde je

$$a = (\sqrt{L}-1)^2, \qquad b = (\sqrt{L}+1)^2. \tag{2.42}$$

Pretpostavimo da važi $x \notin (a,b)$. U ovom slučaju, vrednost $R(x;L)$ je realna. Sada možemo izračunati imaginarni deo izraza $\mathcal{F}(x;L) = \lim_{y\to 0^+}\mathcal{F}(x+iy;L)$ kao

$$\mathrm{Im}\ \mathcal{F}(x;L) = \mathrm{Im}\ [l_2(x)-l_1(x)] = 0.$$

Nasuprot tome, ako važi $x\in(a,b)$, direktnim izračunavanjem (uzimajući u obzir regularne grane korena i logaritma koje smo izabrali) dobijamo sledeće izraze

$$l_1(x) = 2(3L+1)\log\Big[x-(L+1)\pm i\sqrt{4L-(x-L-1)^2}\Big] \tag{2.43}$$

$$\mathrm{Im}\ l_1(x) = \begin{cases} 2(3L+1)\arctan\frac{\sqrt{4L-(x-L-1)^2}}{x-(L+1)}, & x\geq L+1;\\ 2(3L+1)\Big(\pi+\arctan\frac{\sqrt{4L-(x-L-1)^2}}{x-(L+1)}\Big), & x< L+1 \end{cases} \tag{2.44}$$

$$l_2(x) = 2(L-1)\log\left[\frac{-(L-1)^2+2x(L+1)-i(L-1)\sqrt{4L-(x-L-1)^2}}{x^2(L-1)^3}\right] \tag{2.45}$$

$$\mathrm{Im}\ l_2(x) = \begin{cases} 2(L-1)\Big(2\pi+\arctan\frac{x(L+1)-(L-1)^2}{\sqrt{4L-(x-L-1)^2}}\Big), & x\geq\frac{(L-1)^2}{L+1};\\ 2(L-1)\Big(\pi+\arctan\frac{x(L+1)-(L-1)^2}{\sqrt{4L-(x-L-1)^2}}\Big), & x<\frac{(L-1)^2}{L+1} \end{cases} \tag{2.46}$$

Zamenom dobijenih izraza u (2.39), konačno možemo da izračunamo

$$\mathrm{Im}\ \mathcal{F}(x;L) = \lim_{y\to 0^+}\mathrm{Im}\ \mathcal{F}(x+iy;L) = \mathrm{Im}\ l_2(x)-\mathrm{Im}\ l_1(x)-(x-L+1)\sqrt{4L-(x-L-1)^2}$$

Sada izraz (2.37) za težinsku funkciju $\omega(t;L)$ možemo odrediti primenom

$$\omega(x;L) = \psi'(x;L) = -\frac{1}{\pi}\frac{d}{dx}\mathrm{Im}\ \mathcal{F}(x;L) \tag{2.47}$$

i relacije (2.11). $\square$

Napomenimo da formula (2.37) važi samo za $x\in(a,b)$, dok je u suprotnom $\omega(x;L)=0$.

2.6.4 Tročlana rekurentna relacija

Na osnovu Teoreme 2.6.5, težinska funkcija čiji su momenti jednaki $\{a_n(L)\}_{n\in\mathbb{N}_0}$ data je pomoću

$$\omega(x;L)=\begin{cases}\frac{\sqrt{L}}{\pi}\left(1+\frac{1}{x}\right)\sqrt{1-\left(\frac{x-L-1}{2\sqrt{L}}\right)^2}, & x\in((\sqrt{L}-1)^2,(\sqrt{L}+1)^2)\\ 0, & \text{inače}\end{cases} \tag{2.48}$$

Ključna stvar u dokazu naše hipoteze je opisivanje niza polinoma $\{Q_n(x)\}_{n\in\mathbb{N}_0}$ koji su ortogonalni u odnosu na težinu $\omega(x;L)$ zadatu pomoću (2.37) na intervalu (a,b), kao i odredjivanje koeficijenata $\{\alpha_n\}_{n\in\mathbb{N}_0}$ i $\{\beta_n\}_{n\in\mathbb{N}_0}$ tročlane rekurentne relacije.

Primer 2.6.4. Za $L=4$, možemo odrediti prvih nekoliko vrednosti niza polinoma $\{Q_n(x)\}_{n\in\mathbb{N}_0}$. Ove vrednosti su odredjene primenom Gram-Schmidtovog metoda ortogonalizacije i programskog paketa MATHEMATICA.

$$\begin{aligned} Q_0(x)&=1, & \|Q_0\|^2&=5,\\ Q_1(x)&=x-\frac{24}{5}, & \|Q_1\|^2&=\frac{104}{5},\\ Q_2(x)&=x^2-\frac{127}{13}x+\frac{256}{13}, & \|Q_2\|^2&=\frac{1088}{13},\\ Q_3(x)&=x^3-\frac{541}{17}x^2+\frac{1096}{17}x-\frac{1344}{17}, & \|Q_3\|^2&=\frac{5696}{17}.\end{aligned}$$

Takodje važi

$$\alpha_0=\frac{24}{5},\quad \beta_0=5,\qquad \alpha_1=\frac{323}{65},\quad \beta_1=\frac{104}{25},\qquad \alpha_2=\frac{1104}{221},\quad \beta_2=\frac{680}{169}.$$

Prema tome dobijamo

$$h_1(4)=a_0(4)=5,\quad h_2(4)=a_0(4)^2\beta_1=104,\qquad h_3(4)=a_0(4)^3\beta_1^2\beta_2=5^3\left(\frac{104}{25}\right)^2\frac{680}{169}=8704.$$

Da bi odredili koeficijente tročlane rekurentne relacije, primenićemo niz transformacija težinske funkcije. Ove transformacije su definisane sledećim relacijama

$$\begin{aligned} w^*(x)&=p^{(1/2,1/2)}(x)=\sqrt{1-x^2},\\ \hat{w}(x)&=\left(x+\frac{L+2}{2\sqrt{L}}\right)w^*(x),\\ \tilde{w}(x)&=\hat{w}\left(\frac{x}{2\sqrt{L}}-\frac{L+1}{2\sqrt{L}}\right),\\ \breve{w}(x)&=\frac{2L}{\pi}\tilde{w}(x),\\ \omega(x;L)&=\frac{\breve{w}(x)}{x}.\end{aligned}$$

Ostatak ovog odeljka biće posvećen izračunavanju potrebnih koeficijenata tročlane rekurentne relacije primeni predhodno uvedenih transformacionih formula kao i Theoreme 2.5.6, Posledice 2.5.9 i Teoreme 2.5.10.

Transformacija $\hat{w}(x) = \left(x + \frac{L+2}{2\sqrt{L}}\right) w^*(x)$

Na početku, razmotrimo niz moničnih ortogonalnih polinoma $\{S_n(x)\}_{n\in\mathbb{N}_0}$ u odnosu na težinu $w^*(x) = p^{(1/2,1/2)}(x) = \sqrt{1-x^2}$ na intervalu $(-1,1)$. Podsetimo se da su ti polinomi zapravo monični Čebiševljevi polinomi druge vrste definisani sa [17]

$$S_n(x) = \frac{\sin\big((n+1)\arccos x\big)}{2^n\sqrt{1-x^2}}.$$

Oni zadovoljavaju sledeću tročlanu rekurentnu relaciju

$$S_{n+1}(x) = (x-\alpha_n^*)\ S_n(x) - \beta_n^* S_{n-1}(x) \quad (n=0,1,\ldots), \tag{2.49}$$

sa početnim vrednostima

$$S_{-1}(x) = 0, \qquad S_0(x) = 1,$$

gde je

$$\alpha_n^* = 0 \quad (n \geq 0) \qquad \text{i} \qquad \beta_0^* = \frac{\pi}{2}, \quad \beta_n^* = \frac{1}{4} \qquad (n \geq 1).$$

Razmotrimo transformaciju težinske funkcije $\hat{w}(x) = (x-c)w^*(x)$. Na osnovu Posledice 2.5.9, koeficijente $\hat{\alpha}_n$ i $\hat{\beta}_n$ možemo izračunati na sledeći način

$$\begin{aligned} \hat{\alpha}_n &= c - \frac{\lambda_{n+1}}{\lambda_n} - \beta_{n+1}^* \frac{\lambda_n}{\lambda_{n+1}}, \\ \hat{\beta}_n &= \beta_n^* \frac{\lambda_{n-1}\lambda_{n+1}}{\lambda_n^2} \qquad\qquad (n \in \mathbb{N}_0). \end{aligned} \tag{2.50}$$

Niz $\{\lambda_n\}_{n\in\mathbb{N}_0}$ zadovoljava tročlanu rekurentnu relaciju

$$4\lambda_{n+1} - 4c\lambda_n + \lambda_{n-1} = 0, \qquad (\lambda_{-1} = 0;\ \lambda_0 = 1). \tag{2.51}$$

Podsetimo se da važi $\lambda_n = S_n(c)$. Jednačina (2.51) je linearna diferencna jednačina sa konstantnim koeficijentima. Prema tome, možemo napisati njenu karakterističnu jednačinu

$$4z^2 - 4\ c\ z + 1 = 0.$$

Rešenja karakteristične jednačine su

$$z_{1,2} = \frac{1}{2}\Big(c \pm \sqrt{c^2-1}\Big),$$

i na osnovu toga imamo da je opšte rešenje jednačine (2.51) dato sa

$$\lambda_n = E_1 z_1^n + E_2 z_2^n \qquad (n \in \mathbb{N}).$$

Možemo izračunati vrednosti E_1 i E_2 iz početnih uslova $(\lambda_{-1} = 0;\ \lambda_0 = 1)$. Da bi rešili naš problem odabraćemo $c = -\frac{L+2}{2\sqrt{L}}$. Prema tome važi

$$z_k = \frac{-t_k}{4\sqrt{L}} \quad (k=1,2), \quad \text{gdeje} \quad t_{1,2} = L + 2 \pm \sqrt{L^2+4}.$$

Konačno dobijamo sledeći izraz za λ_n

$$\lambda_n = \frac{(-1)^n}{2 \cdot 4^n L^{\frac{n}{2}} \sqrt{L^2+4}} \left(t_1^{n+1} - t_2^{n+1} \right) \quad (n = -1, 0, 1, \dots),$$

Korišćenjem oznaka koje smo uveli u izrazima (2.26) i (2.27) možemo uprostiti predhodni izraz. Tako dobijamo

$$\lambda_n = \frac{(-1)^n}{2 \cdot 4^n L^{\frac{n}{2}} \xi} \, \psi_{n+1} \quad (n = -1, 0, 1, \dots).$$

Nakon zamene u (2.50), konačno odredjujemo potrebne koeficijente tročlane rekurentne relacije

$$\begin{aligned} \hat{\alpha}_n &= -\frac{L+2}{2\sqrt{L}} + \frac{1}{4\sqrt{L}} \cdot \frac{\psi_{n+2}}{\psi_{n+1}} + \sqrt{L} \cdot \frac{\psi_{n+1}}{\psi_{n+2}}, \\ \hat{\beta}_n &= \frac{\psi_n \psi_{n+2}}{4\psi_{n+1}^2}. \end{aligned} \tag{2.52}$$

Transformacija $\tilde{w}(x) = \hat{w}\left(\frac{x}{2\sqrt{L}} - \frac{L+1}{2\sqrt{L}}\right)$

Ukoliko je nova težinska funkcija $\tilde{w}(x)$ zadata pomoću $\tilde{w}(x) = \hat{w}(ax+b)$, na osnovu Teoreme 2.5.6 važi

$$\tilde{\alpha}_n = \frac{\hat{\alpha}_n - b}{a}, \qquad \tilde{\beta}_n = \frac{\hat{\beta}_n}{a^2} \qquad (n \in \mathbb{N}_0).$$

Imamo da je u našem slučaju linearna transformacija definisana pomoću $x \mapsto \frac{x-L-1}{2\sqrt{L}}$. Zamenom $a = \frac{1}{2\sqrt{L}}$ i $b = -\frac{L+1}{2\sqrt{L}}$, dobijamo odgovarajući izraz za novu težinsku funkciju $\tilde{w}(x)$

$$\tilde{w}(x) = \hat{w}(\frac{x-L-1}{2\sqrt{L}}) = \frac{1}{2}\left(\frac{x-L-1}{2\sqrt{L}} + \frac{L+2}{2\sqrt{L}}\right)\sqrt{1 - \left(\frac{x-L-1}{2\sqrt{L}}\right)^2}.$$

Prema tome važi

$$\tilde{\alpha}_n = -1 + \frac{1}{2} \cdot \frac{\psi_{n+2}}{\psi_{n+1}} + 2L \cdot \frac{\psi_{n+1}}{\psi_{n+2}} \qquad (n \in \mathbb{N}_0), \tag{2.53}$$

kao i

$$\tilde{\beta}_0 = (L+2)\frac{\pi}{2}, \qquad \tilde{\beta}_n = L\frac{\psi_n \psi_{n+2}}{\psi_{n+1}^2} \quad (n \in \mathbb{N}). \tag{2.54}$$

Primer 2.6.5. Za $L = 4$, dobijamo

$$\begin{aligned} &\tilde{P}_0(x) = 1, && \|P_0\|^2 = 3\pi, \\ &\tilde{P}_1(x) = x - \frac{17}{3}, && \|P_1\|^2 = \frac{32\pi}{3}, \\ &\tilde{P}_2(x) = x^2 - \frac{43}{4}x + \frac{101}{4}, && \|P_2\|^2 = 42\pi, \\ &\tilde{P}_3(x) = x^3 - \frac{331}{21}x^2 + \frac{1579}{21}x - \frac{2189}{21}, && \|P_3\|^2 = \frac{3520\pi}{21}, \end{aligned}$$

kao i

$$\tilde{\alpha}_0 = \frac{17}{3}, \quad \tilde{\beta}_0 = 3\pi, \qquad \tilde{\alpha}_1 = \frac{61}{12}, \quad \tilde{\beta}_1 = \frac{32}{9}, \qquad \tilde{\alpha}_2 = \frac{421}{84}, \quad \tilde{\beta}_2 = \frac{63}{16}.$$

Transformacija $\breve{w}(x) = \frac{2L}{\pi}\tilde{w}(x)$

Uvodjenjem nove težine $\breve{w}(x) = \frac{2L}{\pi}\tilde{w}(x)$ nećemo promeniti monične ortogonalne polinome i njihovu tročlanu rekurentnu relaciju. Rezultat ove promene je samo promena normi ovih polinoma za faktor $\frac{2L}{\pi}$. Drugim rečima važi

$$\breve{P}_k(x) \equiv P_k(x), \qquad \|\breve{P}_k\|^2_{\breve{w}} = \int_a^b \breve{P}_k(x)\breve{w}(x)\,dx = \frac{2L}{\pi}\|P_k\|^2_{\tilde{w}} \qquad (k \in \mathbb{N}_0),$$

$$\breve{\beta}_0 = L(L+2), \quad \breve{\beta}_k = \tilde{\beta}_k \quad (k \in \mathbb{N}), \qquad \breve{\alpha}_k = \tilde{\alpha}_k \qquad (k \in \mathbb{N}_0).$$

Takodje važi

$$\breve{\beta}_0\breve{\beta}_1 \cdots \breve{\beta}_{n-1} = \frac{L^n}{2} \cdot \frac{\psi_{n+1}}{\psi_n}.$$

Transformacija $\omega(x; L) = \frac{\breve{w}(x)}{x}$

Poslednja transformacija koju ćemo da razmotrimo je oblika

$$w_d(x) = \frac{\breve{w}(x)}{x-d} \qquad (d < \inf \operatorname{supp}(\breve{w})).$$

Definisaćemo pomoćni niz $\{r_n\}_{n\geq -1}$ kao u Teoremi 2.5.10.

$$r_{-1} = -\int_{\mathbb{R}} w_d(x)\,dx, \qquad r_n = d - \breve{\alpha}_n - \frac{\breve{\beta}_n}{r_{n-1}} \quad (n = 0, 1, \ldots), \tag{2.55}$$

Na osnovu Teoreme 2.5.10, koeficijenti $\alpha_{d,n}$ i $\beta_{d,n}$ koji odgovaraju težini $w_d(x)$ mogu biti odredjeni na osnovu sledećih relacija

$$\begin{aligned} \alpha_{d,0} &= \breve{\alpha}_0 + r_0, & \alpha_{d,k} &= \breve{\alpha}_k + r_k - r_{k-1}, & \\ \beta_{d,0} &= -r_{-1}, & \beta_{d,k} &= \breve{\beta}_{k-1}\frac{r_{k-1}}{r_{k-2}} & (k \in \mathbb{N}). \end{aligned} \tag{2.56}$$

U našem slučaju je dovoljno da stavimo $d = 0$ da bi dobili konačan izraz za težinsku funkciju $\omega(x; L) = \frac{\breve{w}(x)}{x}$.

Najpre je potrebno da nadjemo izraz u zatvorenom obliku za niz r_n. Stavljajući $d = 0$ i računajući integral, relacije (2.55) postaju

$$r_{-1} = -(L+1), \qquad r_n = -\Big(\breve{\alpha}_n + \frac{\breve{\beta}_n}{r_{n-1}}\Big) \quad (n = 0, 1, \ldots). \tag{2.57}$$

Rešenje predhodne rekurentne jednačine ćemo intuitivno naslutiti a zatim dokazati matematičkom indukcijom.

Lema 2.6.6. *Niz r_n je zadat sledećim izrazom*

$$r_n = -\frac{\psi_{n+1}}{\psi_{n+2}} \cdot \frac{L\psi_{n+2} + \xi\varphi_{n+2}}{L\psi_{n+1} + \xi\varphi_{n+1}} \qquad (n \in \mathbb{N}_0). \tag{2.58}$$

Dokaz. Koristićemo matematičku indukciju. Za $n = 0$, zamenom u (2.58) zaista dobijamo korektnu vrednost za r_0

$$r_0 = -\frac{L^2 + 2L + 2}{(L+1)(L+2)}.$$

Pretpostavimo da je izraz (2.58) tačan za $k = n$ i dokažimo ga za $k = n + 1$. Sada, na osnovu svojstava funkcija φ_n i ψ_n (Lema 2.6.2), dobijamo

$$\tilde{\alpha}_{n+1} \cdot r_n + \tilde{\beta}_{n+1} = -\frac{\psi_{n+1}}{\psi_{n+3}} \cdot \frac{L\psi_{n+3} + \xi\varphi_{n+3}}{L\psi_{n+1} + \xi\varphi_{n+1}}.$$

Deljenjem sa r_n konačno dobijamo izraz (2.58) za $k = n + 1$. $\square$

Zamenom (2.58) u (2.56) dobijamo izraz u zatvorenom obliku za $\alpha_n = \alpha_{0,n}$ i $\beta_n = \beta_{0,n}$ za svako $n \geq 0$.

Primer 2.6.6. Za $L = 4$ dobijamo

$$r_{-1} = -5, \quad r_0 = -\frac{13}{15}, \quad r_1 = -\frac{51}{52}, \quad r_2 = -\frac{356}{357},$$

odnosno

$$\alpha_0 = \frac{24}{5}, \quad \beta_0 = 5, \qquad \alpha_1 = \frac{323}{65}, \quad \beta_1 = \frac{104}{25}, \qquad \alpha_2 = \frac{1104}{221}, \quad \beta_2 = \frac{680}{169},$$

Primetimo da su ove vrednosti potpuno iste kao u Primeru 2.6.5.

2.6.5 Dokaz glavnog rezultata

Sada smo spremni da dokažemo glavni rezultat ovog poglavlja (Teorema 2.6.1). Neka je $h = \mathbf{H}(a)$. Heilermannovu formulu (2.9) možemo takodje napisati u sledećem obliku

$$h_1(L) = a_0(L), \qquad h_n(L) = \beta_0\beta_1\beta_2 \cdots \beta_{n-2}\beta_{n-1} \cdot h_{n-1}. \tag{2.59}$$

Na osnovu Posledice 2.4.5 važi

$$\|Q_{n-1}\|^2 = \beta_0\beta_1\beta_2 \cdots \beta_{n-2}\beta_{n-1} \quad (n = 2, 3, \ldots), \tag{2.60}$$

odakle dobijamo da je

$$h_1(L) = a_0(L), \qquad h_n(L) = \|Q_{n-1}\|^2 \cdot h_{n-1}(L) \quad (n = 2, 3, \ldots). \tag{2.61}$$

Sada, korišćenjem relacija (2.56) imamo da važi

$$\|Q_{n-1}\|^2 = \beta_0 \frac{r_{n-2}}{r_{-1}} \prod_{k=0}^{n-2} \breve{\beta}_k = \frac{L^{n-1}}{2} \cdot \frac{L\psi_n + \xi\varphi_n}{L\psi_{n-1} + \xi\varphi_{n-1}}. \tag{2.62}$$

Koristićemo matematičku indukciju da bi dokazali da je

$$h_k(L) = \frac{L^{k(k-1)/2}}{2^{k+1}\xi} \cdot (L\psi_k + \xi\varphi_k). \tag{2.63}$$

Poslednji izraz je trivijalno tačan za $k = 1$. Pretpostavimo da važi za $k = n - 1$. Tada je

$$h_n(L) = \frac{L^{n-1}}{2} \cdot \frac{L\psi_n + \xi\varphi_n}{L\psi_{n-1} + \xi\varphi_{n-1}} \cdot \frac{L^{(n-1)(n-2)/2}}{2^n\xi} \cdot (L\psi_{n-1} + \xi\varphi_{n-1}),$$

odakle sledi da izraz (2.63) važi i za $k = n$. Time je dokaz glavnog rezultata ovog poglavlja, Teoreme 2.6.1, kompletiran.

2.7 Hankelova transformacija i k-binomne transformacije

Već smo napomenuli da je Hankelova transformacija invarijantna u odnosu na binomnu transformaciju (Teorema 2.2.2). Ovo tvrdjenje je prvi dokazao Layman u radu [77]. U ovom poglavlju prezentovaćemo kraći i i jednostavniji dokaz ovog tvrdjenja za klasu nizova $\{a_n\}_{n\in\mathbb{N}_0}$ generisanih kao momenti pozitivne težinske funkcije $w(x)$. Takodje razmotrićemo dve generalizacije binomne transformacije (rastuću i opadajuću k-binomnu transformaciju) i uopštićemo naš dokaz na te dve transformacije. Na taj način ćemo dokazati da je Hankelova transformacija invarijantna u odnosu na opadajuću k-binomnu transformaciju. Pored toga, uspostavićemo vezu izmedju Hankelove transformacije originalnog i niza transformisanog k-binomnom transformacijom.

Rezultati prikazani u ovom poglavlju su originalni i bazirani na još uvek neobjavljenom radu [99].

2.7.1 k-Binomne transformacije

Rastuću i opadajuću k-binomnu transformaciju su definisali Spivey i Stail u radu [122]. Podsetimo se da je binomna transformacija niza $\{a_n\}_{n\in\mathbb{N}_0}$, niz $\{b_n\}_{n\in\mathbb{N}_0} = \mathbf{B}(a)$ definisan sa

$$b_n = \sum_{i=0}^{n} \binom{n}{i} a_i. \tag{2.64}$$

Definicija 2.7.1. *Rastuća i opadajuća k-binomna transformacija niza $\{a_n\}_{n\in\mathbb{N}_0}$ su redom nizovi $\{r_n\}_{n\in\mathbb{N}_0} = \mathbf{Br}(a;k)$ i $\{f_n\}_{n\in\mathbb{N}_0} = \mathbf{Bf}(a;k)$ definisani sa*

$$r_n = \sum_{i=0}^{n} \binom{n}{i} k^i a_i; \qquad f_n = \sum_{i=0}^{n} \binom{n}{i} k^{n-i} a_i. \tag{2.65}$$

Jasno je iz (2.65) da važi $\mathbf{B}(a) = \mathbf{Br}(a;1) = \mathbf{Bf}(a;1)$. Prema tome, rastuća i opadajuća k-binomna transformacija predstavljaju uopštenja binomne transformacije.

Teorema 2.7.1. [122] (Spivey, Stail 2006) *Za zadati niz $\{a_n\}_{n\in\mathbb{N}_0}$, neka je $\{h_n\}_{n\in\mathbb{N}_0} = \mathbf{H}(a)$. Opadajuća k-binomna transformacija je invarijantna u odnosu na Hankelovu transformaciju, tj. važi $\mathbf{H}(a) = \mathbf{H}(\mathbf{Bf}(a;k)) = \{h_n\}_{n\in\mathbb{N}_0}$. Takodje važi $\mathbf{H}(\mathbf{Br}(a;k)) = k^{n(n+1)}h_n$.*

2.7.2 Dokaz glavnog rezultata

Dokaz Teoreme 2.7.1 u radu [122] je kombinatorni. Takodje u tom radu je dato nekoliko kombinatornih interpretacija ovih transformacija.

Mi ćemo dati novi dokaz Theoreme 2.7.1 baziran na Heilermannovoj formuli i teoriji ortogonalnih polinoma. Ovaj dokaz je takodje validan za proizvoljni realni broj k što nije slučaj sa dokazom datim u [122].

Podsetimo se da ukoliko su svi elementi Hankelove transformacije niza $\{a_n\}_{n\in\mathbb{N}_0}$ nenegativni, tada postoji pozitivna mera $d\lambda$ tako da je a_n vrednost n-tog momenta mere $d\lambda$, za svako $n \in \mathbb{N}_0$. Ova činjenica važi na osnovu Teoreme 2.5.3 (Hamburgerov momentni problem). Prema tome, naš dokaz je limitiran na slučaj kada su svi elementi Hankelove transformacije polaznog niza nenegativni.

Neka je $w(x)$ odgovarajuća težinska funkcija koja odgovara meri μ. Tada važi

$$a_n = \int_{\mathbb{R}} x^n w(x)dx.$$

Dokaz Teoreme 2.7.1. Uvešćemo sledeće oznake $\{f_n\}_{n\in\mathbb{N}_0} = \mathbf{Bf}(a;k)$ i $\{r_n\}_{n\in\mathbb{N}_0} = \mathbf{Br}(a;k)$. Niz $\{f_n\}_{n\in\mathbb{N}_0}$ je niz n-tih momenata težinske funkcije $w(x-k)$.

$$f_n = \sum_{i=0}^{n} \binom{n}{i} k^{n-i} \int_{\mathbb{R}} x^i w(x)dx = \int_{\mathbb{R}} \left(\sum_{i=0}^{n} \binom{n}{i} k^{n-i} x^i \right) dx = \int_{\mathbb{R}} (x+k)^n w(x)dx \tag{2.66}$$

Uvodjenjem smene promenljivih $x \to x-k$ u predhodnom integralu dobijamo traženi rezultat.

Posmatrajmo sada moničke polinome $P_n(x)$ i $P_n^f(x)$ ortogonalne u odnosu na težine $w(x)$ i $w(x-k)$ respektivno kao i odgovarajuće koeficijente tročlane rekurentne relacije β_n i β_n^f. Na osnovu Teoreme 2.5.6 važi $P_n^f(x) = P_n(x-k)$ kao i $\beta_n = \beta_n^f$. Sada iz Heilermannove formule (2.9) možemo zaključiti da su Hankelove transformacije nizova $\{f_n\}_{n\in\mathbb{N}_0}$ i $\{a_n\}_{n\in\mathbb{N}_0}$ jednake, pošto važi $a_0 = f_0$.

Slično kao u relaciji (2.66), možemo dokazati da je niz $\{r_n\}_{n\in\mathbb{N}_0}$ zapravo niz n-tih momenata težine $w\left(\frac{x-1}{k}\right)$.

$$\begin{aligned} r_n &= \sum_{i=0}^{n} \binom{n}{i} k^{i} \int_{\mathbb{R}} x^i w(x)dx = \int_{\mathbb{R}} \left(\sum_{i=0}^{n} \binom{n}{i} k^{i} x^i \right) dx \\ &= \int_{\mathbb{R}} (1+kx)^n w(x)dx = \int_{\mathbb{R}} x^n w\left(\frac{x-1}{k}\right) dx \end{aligned} \tag{2.67}$$

Primenom Teoreme 2.5.6 dobijamo $\beta_n^r = k^2\beta_n$. Zamenom u Heilermannovu formulu (2.9) možemo direktno zaključiti da važi $\mathbf{H}(r) = \left\{k^{n(n+1)}h_n\right\}_{n\in\mathbb{N}_0}$. $\square$

2.8 Hankelova transformacija niza generalisanih centralnih trinomnih koeficijenata

U ovom poglavlju proučavaćemo generalizaciju niza centralnih trinomnih koeficijenata sa posebnim osvrtom na odredjivanje težinske funkcije čiji su momenti članovi tog niza. Ovi rezultati biće iskorišćeni za izračunavanje Hankelove transformacije posmatranog niza u zatvorenom obliku.

Rezultati prikazani u ovom poglavlju su originalni i bazirani na našem još uvek neobjavljenom radu [8].

2.8.1 Generalisani centralni trinomni koeficijenti

Niz generalisanih centralnih trinomnih koeficijenata definisao je Noe u radu [94]. Neka su a, b, c celi brojevi. Koeficijent uz x^n u razvoju sledećeg polinoma

$$(a + bx + cx^2)^n \tag{2.68}$$

zvaćemo *generalisani centralni trinomni koeficijent* i označavaćemo sa $T_n(a, b, c)$. Ovaj niz je prirodna generalizacija niza centralnih trinomnih koeficijenata $1, 1, 3, 7, 19, 51, \ldots$ A002426, definisanih pomoću $t_n = [x^n](1 + x + x^2)^n$. Ovaj zaključak sledi iz činjenice da važi $t_n = T_n(1, 1, 1)$.

Niz $\{T_n(a, b, c)\}_{n\in\mathbb{N}_0}$, za različite vrednosti a, b, c ima različite kombinatorne interpretacije i pojavljuje se veoma često u Sloaneovoj on-line enciklopediji celih brojeva [121]. Različita svojstva niza $\{T_n(a, b, c)\}_{n\in\mathbb{N}_0}$ data su u [94].

Ako iskoristimo binomnu teoremu dvaput kao i identitet $\binom{n}{k}\binom{n-k}{n-2k} = \binom{2k}{k}\binom{n}{2k}$, dobijamo

$$T_n(a, b, c) = \sum_{k=0}^{\lfloor n/2\rfloor} \binom{2k}{k}\binom{n}{2k} b^{n-2k}(ac)^k. \tag{2.69}$$

Ako pretpostavimo da je b fiksiran broj, primetimo da svi elementi niza $\{T_n(a, b, c)\}_{n\in\mathbb{N}_0}$ zavise isključivo od proizvoda brojeva a i c. Prema tome, možemo pretpostaviti da je $a = 1$. Pošto je $b = 0$ degenerisan slučaj i apsolutna vrednost niza T_n je ista za b i $-b$, možemo pretpostaviti da je $b > 0$.

Kao što ćemo videti u nastavku, važna veličina koja određuje svojstva niza $\{T_n(a, b, c)\}_{n\in\mathbb{N}_0}$ je diskriminanta $d = b^2 - 4ac$. Postoji beskonačno mnogo parova (b, c) za koje je diskriminanta jednaka. Takođe, diskriminanta je uvek broj oblika $4k$ ili $4k + 1$ za neki celi broj k.

Može se pokazati (videti npr. [94]) da je funkcija generatrisa niza $\{T_n(a, b, c)\}_{n\in\mathbb{N}_0}$ jednaka

$$\frac{1}{\sqrt{1 - 2bx + dx^2}} = \sum_{k=0}^{\infty} T_k(a, b, c)x^k. \tag{2.70}$$

Još je Ojler u svom radu [33] pronašao ovu funkciju generatrisu u slučaju $a = b = c = 1$. Takođe sledeća rekurentna jednačina za niz $T_n = T_n(a, b, c)$ važi

$$T_0 = 1, \quad T_1 = b, \quad T_n = \frac{(2n-1)bT_{n-1} - (n-1)dT_{n-2}}{n}, \tag{2.71}$$

U nastavku ćemo izračunati Hankelovu transformaciju niza $\{T_n(1, b, c)\}_{n\in\mathbb{N}_0}$.

2.8.2 Trinomni koeficijenti i moment reprezentacije

Izvešćemo izraz za težinsku funkciju čiji su momenti članovi niza $\{T_n(1, b, c)\}_{n\in\mathbb{N}_0}$. Ovo možemo da uradimo direktnom primenom Stieltjesove inverzne formule kao što je to urađeno u jednom od predhodnih poglavlja. Ovde ćemo koristiti drugačiji pristup. Najpre ćemo odrediti ovu moment reprezentaciju u dva specijalna slučaja a zatim ćemo korišćenjem tog rezultata intuitivno doći do konačnog izraza za moment reprezentaciju u opštem slučaju. Ovako dobijenu moment reprezentaciju ćemo zatim i formalno dokazati.

Počinjemo sa nizom $\{e_n\}_{n\in\mathbb{N}_0}$ isprekidanih centralnih binomnih koeficijenata čiji je opšti član definisan sa

$$e_n = \binom{n}{\lfloor \frac{n}{2} \rfloor} \frac{1+(-1)^n}{2}.$$

Ovaj niz ima sledeću moment reprezentaciju [121]

$$e_n = \frac{1}{\pi} \int_{-2}^{2} \frac{x^n}{\sqrt{4-x^2}} dx. \tag{2.72}$$

Niz $\{t_n\}_{n\in\mathbb{N}_0}$ centralnih trinomnih koeficijenata jednak je binomnoj transformaciji niza $\{e_n\}_{n\in\mathbb{N}_0}$, tj. važi $t = \mathbf{B}(e)$. Ovu činjenicu možemo dokazati na sledeći način

$$t_n = \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k}\binom{2k}{k} = \sum_{k=0}^{n} \binom{n}{k}\binom{k}{\lfloor k/2 \rfloor}\frac{1+(-1)^k}{2} = \sum_{k=0}^{n}\binom{n}{k} e_k$$

Na osnovu dokaza Teoreme 2.7.1, binomna transformacija je ekvivalentna smeni promenljivih u moment reprezentaciji. Drugim rečima važi

$$t_n = \frac{1}{\pi} \int_{-2}^{2} \frac{(1+x)^n}{\sqrt{4-x^2}} dx, \tag{2.73}$$

što je ekvivalentno sa

$$t_n = \frac{1}{\pi} \int_{-1}^{3} \frac{x^n}{\sqrt{3+2x-x^2}} dx. \tag{2.74}$$

Sa druge strane, na osnovu sledećeg identiteta (videti npr. [42] ili [144])

$$\sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k}\binom{2k}{k} 2^{n-2k} = \binom{2n}{n} \tag{2.75}$$

možemo zaključiti da važi $T(1,2,1) = \binom{2n}{n}$. Prema tome, generalisani centralni trinomni koeficijenti predstavljaju generalizaciju centralnih binomnih koeficijenata $\left\{\binom{2n}{n}\right\}_{n\in\mathbb{N}_0}$. Podsetimo se da centralni binomni koeficijenti $1, 2, 6, 20, 70, \ldots$ A000984 imaju sledeću moment reprezentaciju [121]

$$\binom{2n}{n} = \frac{1}{\pi} \int_{0}^{4} \frac{x^n}{\sqrt{x(4-x)}} dx. \tag{2.76}$$

Smenom promenljivih $x \to x-1$ u poslednjem izrazu dobijamo

$$\binom{2n}{n} = \frac{1}{\pi} \int_{-1}^{3} \frac{(1+x)^n}{\sqrt{3+2x-x^2}} dx. \tag{2.77}$$

Odavde možemo da zaključimo da je binomna transformacija niza centralnih trinomnih koeficijenata t_n jednaka nizu centralnih binomnih koeficijenata, tj. $\mathbf{B}(t) = \mathbf{B}\left(\left\{\binom{2n}{n}\right\}_{n\in\mathbb{N}_0}\right)$. Još jedna smena promenljivih nam daje

$$\binom{2n}{n} = \frac{1}{\pi}\int_{-2}^{2}\frac{(2+x)^n}{\sqrt{4-x^2}}dx. \tag{2.78}$$

Prema tome, dobili smo moment reprezentaciju za dva specijalna slučaja niza $T_n(a,b,c)$. Jednačine (2.74) i (2.78) odgovaraju nizovima $T_n(1,1,1)$ i $T_n(1,2,1)$. Prema tome, intuitivno možemo da formulišemo sledeću lemu.

Lema 2.8.1. *Niz*

$$\tau_n(b,c) = \frac{1}{\pi}\int_{-2}^{2}\frac{(b+\sqrt{c}x)^n}{\sqrt{4-x^2}}dx \tag{2.79}$$

jednak je $T_n(1,b,c)$.

Dokaz. Sada možemo skoro pravolinijski izvesti dokaz ove leme. Važe jednakosti

$$\begin{aligned}
\frac{1}{\pi}\int_{-2}^{2}\frac{(b+\sqrt{c}x)^n}{\sqrt{4-x^2}}dx &= \sum_{k=0}^{n}\binom{n}{k}\sqrt{c}^k b^{n-k}\frac{1}{\pi}\int_{-2}^{2}\frac{x^k}{\sqrt{4-x^2}}dx \\
&= \sum_{k=0}^{n}\binom{n}{k}\sqrt{c}^k b^{n-k}\binom{k}{\lfloor\frac{k}{2}\rfloor}(1+(-1)^k)/2 \\
&= \sum_{k=0}^{\lfloor\frac{n}{2}\rfloor}\binom{n}{2k}\binom{2k}{k}\sqrt{c}^{2k}b^{n-2k} = \sum_{k=0}^{\lfloor\frac{n}{2}\rfloor}\binom{n}{2k}\binom{2k}{k}c^k b^{n-2k}.
\end{aligned}$$

□

Napomenimo još jednom da Lemu 2.8.1 možemo da dokažemo korišćenjem funkcije generatrise niza $T_n(1,b,c)$, određene izrazom (2.70) i Stieltjesove inverzione formule, kao što je to urađeno u jednom od predhodnih poglavlja.

Posledica 2.8.2. *Važi sledeća moment reprezentacija*

$$T_n(1,b,c) = \frac{1}{\pi}\int_{-1}^{1}\frac{(b+2\sqrt{c}x)^n}{\sqrt{1-x^2}}dx = \frac{1}{\pi}\int_{b-2\sqrt{c}}^{b+2\sqrt{c}}\frac{y^n}{\sqrt{4c-(y-b)^2}}dy.$$

Dokaz. Direktno iz (2.79), korišćenjem smena promenljivih $x \to 2x$ i $x \to 2(b+2\sqrt{c}x)$. □

2.8.3 Hankelova transformacija niza $T_n(1,\alpha+\beta,\beta^2)$

Sada možemo da izvedemo konačan izraz za Hankelovu transformaciju niza $\{T_n(1,b,c)\}_{n\in\mathbb{N}_0}$. Iz razloga jednostavnosti, uvešćemo sledeću smenu promenljivih $\alpha+\beta = b$ i $\beta^2 = c$.

Propozicija 2.8.3. *Važi*

$$\frac{1}{\pi}\int_{-1}^{3}\frac{(\alpha+\beta x)^n}{\sqrt{3+2x-x^2}}dx = T_n(1,\alpha+\beta,\beta^2).$$

Dokaz. Primenićemo smenu promenljivih $x = y-1$ na integral

$$\frac{1}{\pi}\int_{-2}^{2}\frac{(b+\sqrt{c}x)^n}{\sqrt{4-x^2}}dx.$$

Na taj način dobijamo

$$\frac{1}{\pi}\int_{-1}^{3}\frac{(b-\sqrt{c}+\sqrt{c}y)^n}{\sqrt{3+2y-y^2}}dy = [x^n](1+bx+cx^2)^n.$$

Sada zamenom $\alpha+\beta = b$ i $\beta^2 = c$ dobijamo traženi rezultat. $\square$

Sledeća lema je veoma korisna zato što se pomoću nje svodi računanje Hankelove transformacije niza $\{T_n(1,\alpha+\beta,\beta^2)\}_{n\in\mathbb{N}_0}$ na isti posao samo za mnogo prostiji niz $\{\beta^n t_n\}_{n\in\mathbb{N}_0}$.

Lema 2.8.4. *Niz* $\{T_n(1,\alpha+\beta,\beta^2)\}_{n\in\mathbb{N}_0}$ *je opadajuća* α*-binomna transformacija niza* $\{\beta^n t_n\}_{n\in\mathbb{N}_0}$*, tj. važi* $\mathbf{Bf}\left(\{\beta^n t_n\}_{n\in\mathbb{N}_0};\alpha\right) = \{T_n(1,\alpha+\beta,\beta^2)\}_{n\in\mathbb{N}_0}$.

Dokaz. Koristićemo moment reprezentaciju niza $\{T_n(1,\alpha+\beta,\beta^2)\}_{n\in\mathbb{N}_0}$ (Propozicija 2.8.3) kao i niza $\{\beta^n t_n\}_{n\in\mathbb{N}_0}$ (relacija (2.78)). Imamo da važi

$$\frac{1}{\pi}\int_{-1}^{3}\frac{(\alpha+\beta x)^n}{\sqrt{3+2x-x^2}} = \sum_{k=0}^{n}\binom{n}{k}\alpha^{n-k}\frac{1}{\pi}\int_{-1}^{3}\frac{\beta^k x^k}{\sqrt{3+2x-x^2}}dx = \sum_{k=0}^{n}\binom{n}{k}\alpha^{n-k}\beta^k t_k.$$

Odavde direktno dobijamo da važi uslov leme. $\square$

Za računanje Hankelove transformacije niza $\{t_n\}_{n\in\mathbb{N}_0}$ koristićemo metod baziran na ortogonalnim polinomima. Celokupno izvođenje je znatno prostije nego izvođenje prikazano u jednom od predhodnih poglavlja.

Lema 2.8.5. *Hankelova transformacija niza* $\{t_n\}_{n\in\mathbb{N}_0}$ *je niz* $\{2^n\}_{n\in\mathbb{N}_0}$.

Dokaz. Iz jednačine (2.78) imamo da je niz $\{t_n\}_{n\in\mathbb{N}_0}$ zapravo niz momenata sledeće težinske funkcije

$$w(x) = \frac{1}{\sqrt{3+2x-x^2}} = \frac{1}{2\sqrt{1-(\frac{1-x}{2})^2}}.$$

Težinskoj funkciji $w^*(x) = \frac{1}{\sqrt{1-x^2}}$ odgovaraju monični Čebiševljevi polinomi prve vrste za koje važi

$$\alpha_k^* = 0, \qquad \beta_0^* = \pi, \qquad \beta_1^* = \frac{1}{2}, \qquad \beta_k^* = \frac{1}{4}, \quad (k\in\mathbb{N}).$$

Neka je sada $\tilde{w}(x) = w^*(\frac{1-x}{2}) = w^*(a + bx)$ gde je

$$a = -\frac{1}{2} \quad i \quad b = \frac{1}{2}.$$

Tada važi

$$\begin{aligned}
\tilde{\alpha}_n &= \frac{\alpha_n^* - b}{a} = \frac{0 - \frac{1}{2}}{-\frac{1}{2}} = 1 \\
\tilde{\beta}_0 &= \frac{\beta_0^*}{a} = \frac{\pi}{-\frac{1}{2}} = -2\pi \\
\tilde{\beta}_1 &= \frac{\beta_1^*}{a^2} = \frac{\frac{1}{2}}{\frac{1}{4}} = 2 \\
\tilde{\beta}_k &= \frac{\beta_k^*}{a^2} = \frac{\frac{1}{4}}{\frac{1}{4}} = 1, \qquad k > 1.
\end{aligned}$$

Konačno primenom transformacije $w(x) = \frac{1}{2}\tilde{w}(x) = \frac{1}{\sqrt{3+2x-x^2}}$ dobija se

$$\alpha_n = 1, \quad \beta_0 = -\pi, \quad \beta_1 = 2, \quad \beta_k = 1, \quad k > 1.$$

Odavde direktno primenom Heilermannove formule dobijamo da važi $h_n = 2^n$. $\square$

Sada konačno možemo da damo izraz u zatvorenom obliku za Hankelovu transformaciju niza $\{T_n(1, \alpha + \beta, \beta^2)\}_{n\in\mathbb{N}_0}$.

Teorema 2.8.6. *Hankelova transformacija niza* $\{T_n(1, \alpha + \beta, \beta^2)\}_{n\in\mathbb{N}_0}$ *je niz* $\left\{2^n \beta^{n(n+1)}\right\}_{n\in\mathbb{N}_0}$.

Dokaz. Na osnovu Teoreme 2.7.1, Hankelova transformacija je invarijantna u odnosu na opadajuću α-binomnu transformaciju. Odavde možemo zaključiti da je Hankelova transformacija niza $\{T_n(1, \alpha + \beta, \beta^2)\}_{n\in\mathbb{N}_0}$ jednaka je Hankelovoj transformaciji niza $\beta^n t_n$. Rezultat teoreme sada sledi direktno iz predhodne leme i determinantske definicije Hankelove transformacije. $\square$

Pošto je $c = \beta^2$ imamo sledeću posledicu

Posledica 2.8.7. *Hankelova transformacija niza* $\{T_n(a, b, c)\}_{n\in\mathbb{N}_0}$ *jednaka je* $\left\{2^n (ac)^{\binom{n+1}{2}}\right\}_{n\in\mathbb{N}_0}$.

2.8.4 Generalizacije dobijenih rezultata

1. Glavni rezultat (Theorema 2.8.6) možemo dokazati i malo drugačije. Uvedimo smenu promenljivih $x = y - 1$ u integral

$$T_n(1, \alpha + \beta, \beta^2) = \frac{1}{\pi} \int_{-1}^{3} \frac{(\alpha + \beta x)^n}{\sqrt{3 + 2x - x^2}} dx.$$

Važi $dx = dy$, $\alpha+\beta x = \alpha-\beta+\beta y$ i $3+2x-x^2 = y(4-y)$. Na taj način dobijamo

$$\begin{aligned} T_n(1,\alpha+\beta,\beta^2) &= \frac{1}{\pi}\int_0^4 \frac{(\alpha-\beta+\beta x)^n}{\sqrt{x(4-x)}}dx \\ &= \sum_{k=0}^{n}\binom{n}{k}\beta^k(\alpha-\beta)^{n-k}\frac{1}{\pi}\int_0^4 \frac{x^k}{\sqrt{x(4-x)}}dx \\ &= \sum_{k=0}^{n}\binom{n}{k}\beta^k(\alpha-\beta)^{n-k}\binom{2k}{k}. \end{aligned}$$

Iz poslednje relacije možemo da zaključimo da važi

$$\mathbf{Bf}\left(\left\{\beta^n\binom{2n}{n}\right\}_{n\in\mathbb{N}_0};\alpha-\beta\right) = \left\{T_n(1,\alpha+\beta,\beta^2)\right\}_{n\in\mathbb{N}_0}.$$

Pošto je Hankelova transformacija niza $\binom{2n}{n}$ jednaka 2^n, direktno dobijamo rezultat Teoreme 2.8.6.

2. Posmatrajmo sada novi niz koji se dobija zamenom centralnog binomnog koeficijenta $\binom{2k}{k}$ sa Catalanovim brojem $C_k = \frac{1}{k+1}\binom{2k}{k}$ A000108 u predhodnom izrazu. Novi niz $\{l_n\}_{n\in\mathbb{N}_0}$ ima sledeći opšti član

$$l_n = \sum_{k=0}^{n}\binom{n}{k}\beta^k(\alpha-\beta)^{n-k}C_k.$$

Uvođenjem smene promenljivih $x = y+1$ dobijamo

$$\begin{aligned} l_n &= \frac{1}{2\pi}\int_0^4 \sum_{k=0}^{n}\binom{n}{k}(\alpha-\beta)^{n-k}\beta^k x^k\frac{\sqrt{x(4-x)}}{x}dx \\ &= \frac{1}{2\pi}\int_0^4 (\alpha-\beta+\beta x)^n\frac{\sqrt{x(4-x)}}{x}dx \\ &= \frac{1}{2\pi}\int_{-1}^3 (\alpha+\beta y)^n\frac{\sqrt{3+2y-y^2}}{y+1}dy \\ &= \sum_{k=0}^{n}\binom{n}{k}\beta^k\alpha^{n-k}\tilde{M}(k). \end{aligned} \tag{2.80}$$

Niz $\left\{\tilde{M}(n)\right\}_{n\in\mathbb{N}_0}$ je u literaturi poznat kao niz 'Motzkinovih suma' (A005043, prvih nekoliko članova ovog niza su $1,0,1,1,3,6,15,\ldots$). Ovaj niz predstavlja niz momenta sledeće težinske funkcije

$$w_M(x) = \frac{1}{2\pi}\int_{-1}^3 \frac{\sqrt{3+2x-x^2}}{x+1}dx.$$

Uvođenjem smene promenljivih $x = y-1$ u predhodnom integralu i na osnovu dokaza Teoreme 2.7.1, možemo zaključiti da je $\left\{\tilde{M}(n)\right\}_{n\in\mathbb{N}_0}$ inverzna binomna transformacija niza Catalanovih brojeva $\{C_n\}_{n\in\mathbb{N}_0}$, tj. da važi

$$\mathbf{B}\left(\left\{\tilde{M}(n)\right\}_{n\in\mathbb{N}_0}\right) = \{C_n\}_{n\in\mathbb{N}_0}.$$

Na osnovu Teoreme 2.7.1 oba niza imaju istu Hankelovu transformaciju, tj.

$$\mathbf{H}\left(\left\{\tilde{M}(n)\right\}_{n\in\mathbb{N}_0}\right) = \{1\}_{n\in\mathbb{N}_0}.$$

Iz relacije (2.80) možemo zaključiti da važi

$$\{l_n\}_{n\in\mathbb{N}_0} = \mathbf{Bf}\left(\left\{\beta^k \tilde{M}(k)\right\}_{n\in\mathbb{N}_0}; \alpha\right),$$

tj. niz $\{l_n\}_{n\in\mathbb{N}_0}$ je opadajuća α-binomna transformacija niza $\left\{\beta^k \tilde{M}(k)\right\}_{n\in\mathbb{N}_0}$.

Ponovo primenom Teoreme 2.7.1 zaključujemo da je Hankelova transformacija niza $\{l_n\}_{n\in\mathbb{N}_0}$ jednaka nizu $\left\{\beta^{n(n+1)}\right\}_{n\in\mathbb{N}_0}$.

2.9 Hankelova transformacija inverzije niza generalisanih Fibonaccijevih brojeva

U ovom poglavlju razmotrićemo generalisane Fibonaccijeve brojeve koji generalizuju nekoliko važnih nizova (Fibonaccijevi, Jacobsthalovi i Pellovi brojevi). Posmatraćemo niz koji se dobija inverzijom niza generalisanih Fibonaccijevih brojeva i izračunati njegovu Hankelovu transformaciju.

Rezultati izloženi u ovom poglavlju su originalni i bazirani na našem još uvek neobjavljenom radu [9].

2.9.1 Generalisani Fibonaccijevi brojevi

Pod generalisanim Fibonaccijevim brojevima podrazumevamo rešenje sledeće linearne diferencne jednačine

$$x_0 = 0, \quad x_1 = 1, \qquad x_{n+2} = a x_{n+1} + b x_n \qquad (n \in \mathbb{N}_0;\ a, b \in \mathbb{R}) . \tag{2.81}$$

gde su a, b dati brojevi.

Primer 2.9.1. Niz generalisanih Fibonaccijevih brojeva predstavlja uopštenje sledećih nizova:

(1) Fibonaccijevi brojevi A000045 odgovaraju slučaju $a = b = 1$: $0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ \ldots$;

(2) Jacobsthalovi brojevi A001045 odgovaraju slučaju $a = 1$ i $b = 2$: $0,\ 1,\ 1,\ 3,\ 5,\ 11,\ \ldots$;

(3) Pellovi brojevi A000129 odgovaraju slučaju $a = 2$ i $b = 1$: $0,\ 1,\ 2,\ 5,\ 12,\ 29, \ldots$.

Rešenje jednačine (2.81) očigledno ima sledeću funkciju generatrisu

$$f(x) = \frac{x}{1 - ax - bx^2} \qquad (a, b \in \mathbb{R}) . \tag{2.82}$$

Inverzija niza čija je funkcija generatrisa jednaka $f(x)$ je niz čija je funkcija generatrisa $u = g(x)$ rešenje jednačine $f(u) = x$. Prema tome važi

$$g(x) = \frac{\sqrt{(1+ax)^2 + 4bx^2} - (1+ax)}{2bx} \qquad (a, b \in \mathbb{R}) \ . \tag{2.83}$$

Označićemo sa u_n niz čija je funkcija generatrisa $g(x)$, tj. za koji važi $g(x) = \sum_{n=0}^{\infty} u_n x^n$. Primetimo da $x_1 = 1$ povlači $u_1 = 1$.

Primer 2.9.2. U slučaju Fibonaccijevih brojeva ($a = b = 1$), inverzija ovog niza zadata je pomoću

$$\{u_n\}_{n \in \mathbb{N}_0} : \quad 0,\ 1,\ -1,\ 0,\ 2,\ -3,\ -1, 11,\ -15, \ldots,$$

Ovo je niz A007440 koji ima funkciju generatrisu

$$g(x) = \frac{\sqrt{(1+x)^2 + 4x^2} - (1+x)}{2x} \ .$$

2.9.2 Izračunavanje Hankelove transformacije

Neka je $\tilde{u}_n = u_{n+1}$. Primenićemo Radoux-Junodov metod za računanje Hankelove transformacije niza $\{\tilde{u}_n\}_{n \in \mathbb{N}_0}$. Tada važi

$$\Phi(x) = \sum_{n=0}^{\infty} \tilde{u}_n x^n = \frac{2}{1 + ax + \sqrt{(1+ax)^2 + 4bx^2}} \ . \tag{2.84}$$

Definišimo funkciju $\Psi(x)$ pomoću

$$\Psi(x) = 1 - \frac{1}{\Phi(x)} = \frac{1 - ax - \sqrt{(1+ax)^2 + 4bx^2}}{2} \ . \tag{2.85}$$

Očigledno je $\Psi(0) = 0$ kao i $\Psi'(0) = -a$. Definišimo funkciju $\zeta(x)$ na sledeći način

$$\zeta(x) = \frac{\Psi(x)}{x} - \Psi'(0) = \frac{1 + ax - \sqrt{(1+ax)^2 + 4bx^2}}{2} \ . \tag{2.86}$$

Izvršićemo sledeću dekompoziciju

$$\frac{\zeta(x)}{x} = \lambda + \mu\ \zeta(x) + \nu\ \zeta^2(x) \ . \tag{2.87}$$

Možemo zaključiti da su koeficijenti u predhodnom razlaganju jednaki $\lambda = -b$, $\mu = -a$ i $\nu = 1$. Prema tome imamo da je

$$\frac{\zeta(x)}{x} = -b - a\ \zeta(x) + \zeta^2(x) \ . \tag{2.88}$$

Sada možemo direktno da primenimo Teoremu 2.3.5. Ova primena nas konačno dovodi do izraza za Hankelovu transformaciju niza $\{\tilde{u}_n\}_{n \in \mathbb{N}_0}$ u zatvorenom obliku

$$h_n = \lambda^{n(n+1)/2} \cdot \nu^{n(n-1)/2} = (-b)^{n(n+1)/2}. \tag{2.89}$$

Glava 3

Generalisani inverzi konstantnih matrica

U ovoj glavi ćemo dati definicije i osnovna svojstva generalisanih inverza matrica kao i neke od metoda za njihovo izračunavanje. Većina metoda je formulisana u obliku algoritma. Za svaki algoritam je određena vremenska složenost.

Većina rezultata prikazanih u ovoj glavi je preuzeta iz literature, osim poslednjeg poglavlja koje opisuje novi metod za brzo računanje Moore-Penroseovog i nekih $\{i, j, \ldots, k\}$ inverza. Poslednje poglavlje je bazirano na našem, još uvek neobjavljenom radu [105].

3.1 Definicije i osnovna svojstva

U ovom poglavlju definisaćemo nekoliko klasa generalisanih inverza i proučiti njihova osnovna svojstva. Zbog jednostavnijeg referenciranja, u ovom poglavlju su navedene činjenice, definicije i oznake koje ćemo koristiti u ovoj i sledećoj glavi.

3.1.1 Neke oznake i osnovni pojmovi

Označimo sa $\mathbb{R}$ odnosno $\mathbb{C}$ skup realnih odnosno kompleksnih brojeva. Sa $\mathbb{F}$ ćemo označavati proizvoljno polje (iako ćemo u ovom radu koristiti isključivo polja $\mathbb{R}$ i $\mathbb{C}$, sve definicije, tvrđenja, itd... koja sadrže oznaku $\mathbb{F}$ važiće za proizvoljno polje). Neka je $\mathbb{F}^{m\times n}$ skup svih matrica formata (dimenzija) $m \times n$ nad poljem $\mathbb{F}$. Jediničnu matricu formata $n \times n$ označićemo sa I_n dok ćemo dijagonalnu matricu čiji su elementi na glavnoj dijagonali $d_1, d_2, \ldots, d_n$ označiti sa $\mathrm{diag}(d_1, d_2, \ldots, d_n)$. Nula matricu proizvoljnog formata označićemo sa $\mathbb{O}$. Matrica A^* predstavljaće konjugovano-transponovanu matricu matrici A.

U nastavku ćemo formulisati nekoliko osnovnih definicija i teorema iz linearne algebre koje ćemo koristiti u nastavku ovog rada. Ova teorija je preuzeta iz obimne i dobro poznate literature iz oblasti linearne algebre i teorije matrica (na primer [137, 83, 160]).

Najpre ćemo definisati nekoliko osnovnih klasa matrica.

Definicija 3.1.1. *Kvadratna matrica* $A \in \mathbb{C}^{n\times n}$ $(A \in \mathbb{R}^{n\times n})$ *je*

- *Hermitska (simetrična) ako važi* $A^* = A$ $(A^T = A)$,
- *normalna ako važi* $A^*A = AA^*$ $(A^TA = AA^T)$,
- *unitarna (ortogonalna) ako važi* $A^* = A^{-1}$ $(A^T = A^{-1})$
- *donje trougaona ako važi* $a_{ij} = 0$ *za* $i > j$,
- *gornje trougaona ako važi* $a_{ij} = 0$ *za* $i < j$.
- *pozitivno semi-definitna ako je* $\mathrm{Re}\,(x^*Ax) \geq 0$ *za svako* $x \in \mathbb{C}^{n\times 1}$. *Ako još važi* $\mathrm{Re}\,(x^*Ax) > 0$ *za svako* $x \in \mathbb{C}^{n\times 1} \setminus \{\mathbb{O}\}$, *matrica* A *je pozitivno definitna.*

Sledeća lema daje jedno važno svojstvo Hermitskih, pozitivno definitnih matrica.

Lema 3.1.1. *Neka je* $A \in \mathbb{C}^{n\times n}$ *Hermitska, pozitivno definitna matrica. Tada postoji matrica* $B \in \mathbb{C}^{n\times n}$, *takva da je* $A = B^*B$. *Matrica* B *se naziva kvadratni koren matrice* A *i označava sa* $B = A^{1/2}$.

Definicija 3.1.2. *Neka je* $A \in \mathbb{C}^{m\times n}$ *proizvoljna matrica. Definisaćemo jezgro matrice* A, *u oznaci* $N(A)$, *kao inverznu sliku nula vektora, tj.*

$$N(A) = \{x \in \mathbb{C}^{n\times 1} \mid Au = \mathbb{O}\}.$$

Takodje, definisaćemo sliku matrice A, *u oznaci* $R(A)$, *kao skup svih slika svih vektora, tj.*

$$R(A) = \{y \in \mathbb{C}^{m\times 1} \mid y = Ax \textit{ for some } x \in \mathbb{C}^{n\times 1}\}.$$

Dimenzija slike $R(A)$ naziva se *rang* matrice A i obeležava sa $\mathrm{rank}A$. Takodje, sa $\mathbb{F}_r^{m\times n}$ označićemo skup svih matrica formata $m \times n$ nad poljem $\mathbb{F}$ čiji je rang jednak r.

Osim ranga, druga važna karakteristika svake matrice je njen indeks.

Propozicija 3.1.2. *Za svaku matricu* $A \in \mathbb{C}^{n\times n}$ *postoji prirodan broj* k *takav da je* $\mathrm{rank}A^{k+1} = \mathrm{rank}A^k$.

Definicija 3.1.3. *Neka je* $A \in \mathbb{C}^{n\times n}$ *proizvoljna matrica. Najmanji ceo broj* k *takav da važi* $\mathrm{rank}(A^{k+1}) = \mathrm{rank}(A^k)$ *naziva se indeks matrice* A *i označava sa* $\mathrm{ind}A = k$.

Primetimo da ako je A regularna, tada je $\mathrm{ind}A = 0$ dok je u suprotnom $\mathrm{ind}A \geq 1$. Indeks matrice ima veliku ulogu u proučavanju Drazinovog inverza.

Sledeće poznate dekompozicije (faktorizacije) matrica koristićemo u daljem izlaganju.

Lema 3.1.3. **[137]** *Za svaku matricu* $A \in \mathbb{C}^{m\times n}$ *postoje unitarne matrice* $Q \in \mathbb{C}^{m\times m}$ *i* $P \in \mathbb{C}^{n\times n}$ *takve da važi* $A = Q^*RP$ *gde matrica* R *ima sledeći oblik*

$$R = \begin{bmatrix} R_{11} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} \in \mathbb{C}^{m\times n}, \quad R_{11} \in \mathbb{C}_k^{k\times k}. \tag{3.1}$$

Specijalan slučaj dekompozicije date u Lemi 3.1.3 je poznata singularno-vrednosna dekompozicija (SVD, singular value decomposition) data u sledećoj teoremi.

Teorema 3.1.4. (Singularno-vrednosna dekompozicija) *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica. Postoje unitarne matrice $U \in \mathbb{C}^{m\times m}$ i $V \in \mathbb{C}^{n\times n}$ kao i matrica*

$$\Sigma = \begin{bmatrix} diag(\sigma_1, \sigma_2, \ldots, \sigma_k) & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} \in \mathbb{C}^{m\times n},$$

tako da važi $A = U^*\Sigma V$. Ova dekompozicija se naziva singularno-vrednosna dekompozicija matrice A.

Druge dve važne dekompozicije su LU dekompozicija i Cholesky faktorizacija. One su date u sledećoj teoremi.

Teorema 3.1.5. (LU i Cholesky faktorizacija) *Za svaku regularnu kvadratnu matricu $A \in \mathbb{C}^{n\times n}$ postoji donje trougaona matrica L i gornje trougaona matrica U tako da važi $A = LU$ kao i $l_{ii} = 1$ za svako $i = 1, 2, \ldots, n$. Ova dekompozicija se naziva LU dekompozicija. Štaviše, ako je A Hermitska i pozitivno definitna matrica, tada važi $U = L^*$ a dekompozicija $A = LL^*$ se naziva Cholesky dekompozicija (faktorizacija).*

Pomenućemo takođe i faktorizacije potpunog ranga.

Definicija 3.1.4. (Faktorizacija potpunog ranga) *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica. Faktorizacija $A = FG$, gde je F matrica potpunog ranga kolona (rang ove matrice jednak je broju kolona) a G matrica potpunog ranga vrsta (rang ove matrice jednak je broju vrsta), naziva se faktorizacija potpunog ranga matrice A.*

Teorema 3.1.6. *Neka je $A \in \mathbb{C}^{m\times n}$ takva da je A niti potpunog ranga kolona, niti potpunog ranga vrsta. Tada postoji bar jedna faktorizacija potpunog ranga $A = FG$ matrice A.*

Na kraju ovog odeljka navešćemo čuvenu Jordanovu normalnu formu matrice.

Teorema 3.1.7. (Jordanova normalna forma) *Neka je $A \in \mathbb{C}^{n\times n}$ proizvoljna matrica. Tada postoji regularna matrica P takva da važi $A = PJP^{-1}$ gde je $J = diag(J_1, \ldots, J_k)$ a svaki blok J_i ima oblik*

$$J_i = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \lambda & 1 \\ 0 & 0 & 0 & \cdots & 0 & \lambda \end{bmatrix}.$$

Pritom je λ sopstvena vrednost matrice A.

3.1.2 Moore-Penroseov i $\{i, j, \ldots, k\}$ inverzi

Sledeću teoremu je dokazao Penrose 1955. godine i ona se najčešće uzima kao definicija Moore-Penroseovog (MP) inverza.

Teorema 3.1.8. **[97] (Penrose 1955)** *Za svaku matricu $A \in \mathbb{R}^{m\times n}$, sistem matričnih jednačina*

$$\begin{array}{ll} (1)\ AXA = A, & (3)\ (AX)^T = AX, \\ (2)\ XAX = X, & (4)\ (XA)^T = XA. \end{array} \tag{3.2}$$

ima jedinstveno rešenje $X \in \mathbb{R}^{n\times m}$. Ovo rešenje je poznato kao Moore-Penroseov inverz (generalisani inverz, uopšteni inverz, pseudoinverz) matrice A i označava se sa $A^\dagger$.

Ako je A kvadratna regularna matrica, tada njena inverzna matrica A^{-1} trivijalno zadovoljava sistem (3.2). Iz predhodne teoreme sledi da je MP inverz regularne kvadratne matrice jednak njenoj inverznoj matrici, tj. $A^\dagger = A^{-1}$. Jednačine (3.2) se nazivaju *Penroseove jednačine* i koriste se za definiciju drugih klasa generalisanih inverza.

Definicija 3.1.5. *Neka je $A\{i, j, \ldots, k\}$ skup matrica koje zadovoljavaju jednačine $(i), (j), \ldots, (k)$ među jednačinama $(1), \ldots, (4)$ iz (3.2). Ovakve matrice nazivaju se $\{i, j, \ldots, k\}$ inverzi i označavaju sa $A^{(i,j,\ldots,k)}$.*

Jedno od glavnih svojstava MP inverza je karakterizacija minimalnog srednjekvadratnog rešenja sistema linearnih jednačina $Ax = b$. Ovu karakterizaciju je dokazao Penrose 1955. godine u svom radu [97].

Definicija 3.1.6. *Srednjekvadratno rešenje sistema linearnih jednačina $Ax = b$ gde je $A \in \mathbb{C}^{m\times n}$ i $b \in \mathbb{C}^{m\times 1}$ je vektor $u \in \mathbb{C}^{n\times 1}$ takav da za svako $v \in \mathbb{C}^{n\times 1}$ važi $\|Au - b\| \leq \|Av - b\|$. Kažemo da je srednjekvadratno rešenje u minimalno ako za svako drugo srednjekvadratno rešenje u' važi $\|u\| \leq \|u'\|$.*

Teorema 3.1.9. **[97] (Penrose 1955)** *Neka je $A \in \mathbb{C}^{m\times n}$ i neka je $b \in \mathbb{C}^{m\times 1}$. Minimalno srednjekvadratno rešenje sistema $Ax = b$ dato je pomoću $x^* = A^\dagger b$. Sva ostala srednjekvadratna rešenja data su sledećim izrazom*

$$x = A^\dagger b + (I_n - A^\dagger A)z, \quad z \in \mathbb{C}^{n\times 1}.$$

Primetimo da ukoliko je sistem $Ax = b$ konzistentan (tj ima rešenje) tada je svako srednjekvadratno rešenje ovog sistema ujedno i njegovo rešenje. Važi i obratno. Za konzistentne linearne sisteme $Ax = b$, minimalna (srednjekvadratna) rešenja su karakterisana sledećom teoremom.

Teorema 3.1.10. **[10, 108, 148]** *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica. Tada važi: Za svako $b \in R(A)$, $AXb = b$ i $\|Xb\| < \|u\|$ za svako drugo rešenje $u \neq Xb$ sistema $Ax = b$, ako i samo ako je $X \in A\{1, 4\}$.*

Nasuprot tome, sva srednjekvadratna rešenja nekonzistentnog sistema $Ax = b$ karakterisana su sledećom teoremom.

Teorema 3.1.11. [**10**, **148**] *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica. Tada važi: Xb je srednjekvadratno rešenje sistema $Ax = b$ za svako $b \notin R(A)$, ako i samo ako je $X \in A\{1,3\}$.*

Sledećom lemom su pobrojana osnovna svojstva MP i $\{1\}$ inverza

Lema 3.1.12. *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica. Tada je*

(1) $(A^\dagger)^\dagger = A$, $(A^\dagger)^* = (A^*)^\dagger$;

(2) $(\lambda A)^\dagger = \lambda^\dagger A^\dagger$, *gde je* $\lambda \in \mathbb{C}$ *i* $\lambda^\dagger = \begin{cases} \frac{1}{\lambda}, & \lambda \neq 0 \\ 0, & \lambda = 0 \end{cases}$;

(3) $(AA^*)^\dagger = (A^*)^\dagger A^\dagger$, $(A^*A)^\dagger = A^\dagger (A^*)^\dagger$;

(4) $A^\dagger AA^* = A^* = A^*AA^\dagger$;

(5) $A^\dagger = (A^*A)^\dagger A^* = A^*(AA^*)^\dagger$;

(6) $N(AA^\dagger) = N(A^\dagger) = N(A^*) = R(A)$

(7) $R(AA^*) = R(AA^{(1)}) = R(A)$, $\mathrm{rank}(AA^{(1)}) = \mathrm{rank}(A^{(1)}A) = \mathrm{rank}A$;

(8) *Ako je* $\mathrm{rank}A = m$ *tada je matrica* A^*A *regularna i važi* $A^\dagger = (A^*A)^{-1}A^*$.

Napomenimo da se još neka svojstva $A^\dagger$ i $A^{(1)}$ koja nisu spomenuta u Lemi 3.1.12 mogu naći na primer u [10, 148].

U sledeće tri leme i teoreme pokazaćemo reprezentacije $A^\dagger$ korišćenjem dekompozicija datih u Lemi 3.1.3, SVD dekompozicije (Teorema 3.1.4) kao i faktorizacija potpunog ranga.

Lema 3.1.13. *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica. Posmatrajmo dekompoziciju $A = Q^*RP$ datu u Lemi 3.1.3. Tada se MP inverzi matrica R i A, $R^\dagger$ i $A^\dagger$ mogu napisati na sledeći način*

$$R^\dagger = \begin{bmatrix} R_{11}^{-1} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix}, \quad A^\dagger = Q^*R^\dagger P. \tag{3.3}$$

Kao specijalan slučaj predhodne leme imamo reprezentaciju $A^\dagger$ pomoću SVD dekompozicije matrice A.

Teorema 3.1.14. *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica i neka je $A = U\Sigma V^*$ SVD dekompozicija matrice A. Ako je*

$$\Sigma = \begin{bmatrix} diag(\sigma_1, \sigma_2, \dots, \sigma_k) & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} \in \mathbb{C}^{m\times n},$$

onda važi

$$\Sigma^\dagger = \begin{bmatrix} diag(1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_k) & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} \in \mathbb{C}^{n\times m}$$

i $A^\dagger = V\Sigma^\dagger U^*$.

Teorema 3.1.15. **[84] (MacDuffe, 1956)** *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica i $A = FG$ faktorizacija potpunog ranga matrice A. Tada je*

$$A^\dagger = G^*(F^*AG^*)^{-1}F^* = G^*(GG^*)^{-1}(F^*F)^{-1}F^*.$$

Sledeća teorema predstavlja jedan pomoćni rezultat koji ćemo koristiti u odeljku 4.7.2 naredne glave prilikom razmatranja problema projektovanja povratne sprege. Koliko nam je poznato ovaj rezultat je originalan i zato ovde dajemo kompletan dokaz.

Teorema 3.1.16. *Neka su $A, B \in \mathbb{C}^{m\times n}$ proizvoljne matrice. Tada važi $A = AB^{(1)}B$ i $B = AA^{(1)}B$ ako i samo ako važi $R(A) = R(B)$ i $R(A^*) = R(B^*)$.*

Dokaz.
($\Leftarrow$:) Iz Leme 3.1.12 imamo da važi $\mathrm{rank}A = \mathrm{rank}(AB^{(1)}B) \le \mathrm{rank}(B^{(1)}B) = \mathrm{rank}B$. Na sličan način (korišćenjem $B = AA^{(1)}B$) dobijamo da važi $\mathrm{rank}B \le \mathrm{rank}A$. Ovim smo dokazali da je $\mathrm{rank}A = \mathrm{rank}B$. Sa druge strane imamo da je $R(B) = R(AA^{(1)}B) \subseteq R(AA^{(1)}) = R(A)$ (Lema 3.1.12) i pošto je $\mathrm{rank}B = \mathrm{rank}A$ važi $R(A) = R(B)$. Analogno korišćenjem $A^* = (B^{(1)}B)^*A^*$ i $R((B^{(1)}B)^*) = R(B^*)$ (Lema 3.1.12) dokazujemo da je $R(A^*) = R(B^*)$.
($\Rightarrow$:) Neka je $x \in \mathbb{C}^{n\times 1}$ proizvoljan vektor. Pošto je $R(A) = R(B)$ mora da postoji vektor $y \in \mathbb{C}^{n\times 1}$ takav da je $Bx = Ay$. Prema tome imamo da važi $AA^{(1)}Bx = AA^{(1)}Ay = Ay = Bx$. Ovim smo dokazali relaciju $B = AA^{(1)}B$. Relacija $A = AB^{(1)}B$ se dokazuje potpuno analogno.
□

Sada ćemo pokazati kako se $\{1\}$ inverzi mogu primeniti u rešavanju (tj. karakterizaciji rešenja) nekih matričnih jednačina.

Teorema 3.1.17. **[97] (Penrose 1955)** *Neka je $A \in \mathbb{C}^{m\times n}, B \in C^{p\times q}$ i neka je $D \in \mathbb{C}^{m\times q}$. Tada je matrična jednačina*

$$AXB = D \tag{3.4}$$

konzistentna ako i samo ako za proizvoljne $A^{(1)}$ i $B^{(1)}$ važi

$$AA^{(1)}DB^{(1)}B = D. \tag{3.5}$$

U tom slučaju, opšte rešenje je dato pomoću

$$X = A^{(1)}DB^{(1)} + Y - A^{(1)}AYBB^{(1)}. \tag{3.6}$$

Sledeću posledicu dobijamo direktno iz Teoreme 3.1.17 kada stavimo $B = I_p$.

Posledica 3.1.18. *Neka je $A \in \mathbb{C}^{m\times n}, D \in \mathbb{C}^{m\times p}$. Matrična jednačina $AX = D$ ima rešenja ako i samo ako važi $AA^{(1)}D = D$ za svaki $\{1\}$ inverz $A^{(1)}$ matrice A.*

Sledećom teoremom rešava se sistem matričnih jednačina.

Teorema 3.1.19. *Neka je* $A \in \mathbb{C}^{p\times m}$, $B \in \mathbb{C}^{p\times n}$, $D \in \mathbb{C}^{n\times q}$ *i* $E \in \mathbb{C}^{m\times q}$. *Matrične jednačine*

$$AX = B, \quad XD = E$$

imaju zajedničko rešenje ako i samo ako svaka jednačina posebno ima rešenja i važi $AE = BD$. *U tom slučaju opšte zajedničko rešenje ovih jednačina dato je pomoću*

$$X = X_0 + (I_m - A^{(1)}A)Y(I_n - DD^{(1)})$$

gde su $A^{(1)}$ *i* $D^{(1)}$ *proizvoljni* $\{1\}$ *inverzi,* $Y \in \mathbb{C}^{m\times n}$ *proizvoljna matrica i*

$$X_0 = A^{(1)}B + ED^{(1)} - A^{(1)}AED^{(1)}.$$

3.1.3 Težinski Moore-Penroseov inverz

U predhodnom odeljku diskutovali smo vezu između $A^{(1,4)}$, $A^{(1,3)}$ i $A^\dagger$ sa jedne, i minimalnog rešenja, srednjekvadratnog rešenja kao i minimalnog srednjekvadratnog rešenja linearnog sistema $Ax = b$ (Teorema 3.1.9, Teorema 3.1.10 i Teorema 3.1.11), sa druge strane.

U tim slučajevima razmatrali smo minimizaciju u odnosu na uobičajeni skalarni proizvod $(x, y) = y^*x$ i normu $\|x\| = (x, x)^{1/2}$.

U ovom odeljku ćemo razmotriti težinske norme, kako za rešenje x, tako i za ostatak $Ax - b$ linearnog sistema $Ax = b$.

Neka je $A \in \mathbb{C}^{m\times n}$ i neka su $M \in \mathbb{C}^{m\times m}$ i $N \in \mathbb{C}^{n\times n}$ Hermitske, pozitivno definitne matrice. Težinski skalarni proizvod u prostorima $\mathbb{C}^m$ i $\mathbb{C}^n$, respektivno, možemo definisati na sledeći način

$$(x, y)_M = y^*Mx, \quad (x, y)_N = y^*Nx.$$

Ovako definisani skalarni proizvodi indukuju norme $\|x\|_M$ i $\|x\|_N$ na uobičajen način. Podsetimo se da konjugovano transponovana matrica A^* zadovoljava $(Ax, y) = (x, A^*y)$ za svako $x \in \mathbb{C}^{m\times 1}$ i $y \in \mathbb{C}^{n\times 1}$. Na isti način možemo definisati *težinsku konjugovano transponovanu matricu* $A^{\#}$ koja zadovoljava $(Ax, y)_M = (x, A^{\#})_N$. Sledeća lema dokazuje postojanje i daje izraz za računanje matrice $A^{\#}$.

Lema 3.1.20. *Za svake dve Hermitske, pozitivno definitne matrice* $M \in \mathbb{C}^{m\times m}$ *i* $N \in \mathbb{C}^{n\times n}$ *i proizvoljnu matricu* $A \in \mathbb{C}^{m\times n}$, *postoji jedinstvena matrica* $A^{\#}$ *koja zadovoljava* $(Ax, y)_M = (x, A^{\#})_N$ *i važi* $A^{\#} = N^{-1}A^*M$.

Veze između težinskog MP inverza i rešenja sistema linearnih jednačina date su u nastavku ovog odeljka.

Teorema 3.1.21. *Neka je* $A \in \mathbb{C}^{m\times n}$ *i* N *Hermitska, pozitivno definitna matrica reda* n. *Tada je* $x = Xb$ *minimalno (u odnosu na normu* $\| \ \|_N$*) rešenje konzistentnog sistema linearnih jednačina* $Ax = b$ *za svako* $b \in R(A)$ *ako i samo ako* X *zadovoljava sledeće matrične jednačine*

$$(1)\ AXA = A; \quad (4N)\ (NXA)^* = NXA. \tag{3.7}$$

Svaka matrica X koja je rešenje sistema matričnih jednačina (3.7) naziva se $\{1, 4N\}$ inverz i označava sa $A^{(1,4N)}$. Skup svih $A^{(1,4N)}$ inverza označićemo sa $A\{(1, 4N)\}$.

Teorema 3.1.22. *Neka je $A \in \mathbb{C}^{m\times n}$ i M Hermitska, pozitivno definitna matrica reda m. Tada je $x = Xb$ srednjekvadratno rešenje (u odnosu na normu $\| \, \|_M$) nekonzistentnog sistema linearnih jednačina $Ax = b$ za svako $b \notin R(A)$ ako i samo ako X zadovoljava sledeći sistem matričnih jednačina*

$$(1)\ AXA = A; \quad (3M) (MAX)^* = MAX. \tag{3.8}$$

Svaka matrica X koja zadovoljava (3.8) naziva se $\{1, 3M\}$ inverz i označava sa $A^{(1,3M)}$. Skup svih $A^{(1,3M)}$ inverza označićemo sa $A\{(1, 3M)\}$.

Teorema 3.1.23. *Neka je $A \in \mathbb{C}^{m\times n}$ i neka su M i N Hermitske, pozitivno definitne matrice reda m i n, respektivno. Tada je $x = Xb$ minimalno (u odnosu na normu $\| \, \|_N$) srednjekvadratno rešenje (u odnosu na normu $\| \, \|_M$) nekonzistentnog sistema linearnih jednačina $Ax = b$ za svako $b \notin R(A)$ ako i samo ako X zadovoljava sledeće matrične jednačine*

$$\begin{array}{ll} (1)\ AXA = A, & (3M)\ (MAX)^* = MAX, \\ (2)\ XAX = X, & (4N)\ (NXA)^* = NXA. \end{array} \tag{3.9}$$

Štaviše, sistem matričnih jednačina (3.9) ima jedinstveno rešenje.

Matrica X koja zadovoljava (3.9) naziva se *težinski Moore-Penroseov inverz* i označava sa $X = A^\dagger_{MN}$. Težinski Moore-Penroseov inverz $A^\dagger_{MN}$ je uopštenje Moore-Penroseovog inverza $A^\dagger$. Ako je $M = I_m$, $N = I_n$, onda je $A^\dagger_{MN} = A^\dagger$.

U sledećoj lemi su data neka najosnovnija svojstva težinskog MP inverza.

Lema 3.1.24. *Neka je $A \in \mathbb{C}^{m\times n}$. Ako su $M \in \mathbb{C}^{m\times m}$ i $N \in \mathbb{C}^{n\times n}$ Hermitske, pozitivno definitne matrice, onda važi*

(1) $(A^\dagger_{MN})^\dagger_{NM} = A$,

(2) $(A^\dagger_{MN})^* = (A^*)^\dagger_{N^{-1}M^{-1}}$,

(3) $A^\dagger MN = (A^*MA)^\dagger_{I_m N} A^*M = N^{-1}A^*(AN^{-1}A^*)^\dagger_{MI_n}$,

(4) *Ako je $A = FG$ faktorizacija potpunog ranga matrice A, tada je*

$$A^\dagger_{MN} = N^{-1}G^*(F^*MAN^{-1}G^*)^{-1}F^*M,$$

(5) $A^\dagger_{MN} = N^{-1/2}(M^{-1/2}AN^{-1/2})^\dagger M^{-1/2}$.

3.1.4 Drazinov inverz

U predhodna dva odeljka posmatrali smo MP inverz i ostale $\{i, j, \dots, k\}$ inverze koji imaju odredjena svojstva inverzne matrice. Inverzi tipa $\{i, j, \dots, k\}$ daju neka rešenja, ili srednjekvadratna rešenja linearnih sistema jednačina baš kao što to čini inverzna matrica, u slučaju da postoji. Zato $\{i, j, \dots, k\}$ inverze nazivamo inverzima koji rešavaju jednačine.

Sa druge strane, neka druga svojstva inverzne matrice $\{i, j, \dots, k\}$ inverzi ne poseduju. Na primer $A^- A = AA^-$, $(A^-)^p = (A^p)^-$, itd. Drazinov inverz zadovoljava ova svojstva. Nažalost, ovaj generalisani inverz definisan je samo za kvadratne matrice, ali postoje neka uopštenja i za pravougaone (npr. [10]). Ovaj inverz je nazvan po M. Drazinu koji ga je uveo 1958. godine u radu [28].

Teorema 3.1.25. [28] (Drazin 1958) *Neka je $A \in \mathbb{C}^{n\times n}$ proizvoljna matrica i $k = \mathrm{ind}A$. Tada sledeći sistem matričnih jednačina*

$$(1^k)\ A^k X A = A^k, \quad (2)\ XAX = X, \quad (5)\ AX = XA, \tag{3.10}$$

ima jedinstveno rešenje. Ovo rešenje se naziva Drazinov inverz matrice A i označava sa A^D.

Sledeća lema sadrži osnovna svojstva Drazinovog inverza.

Lema 3.1.26. *Neka je $A \in \mathbb{C}^{n\times n}$ i $k = \mathrm{ind}A$. Tada važi*

(1) $(A^*)^D = (A^D)^*$, $(A^T)^D = (A^D)^T$, $(A^n)^D = (A^D)^n$ *za svako* $n = 1, 2, \dots$,

(2) $((A^D)^D)^D = A^D$, $(A^D)^D = A$ *ako i samo ako je* $k = 1$,

(3) $R(A^D) = R(A^l)$ *i* $N(A^D = N(A^l)$ *za svako* $l \geq k$,

(4) *Ako je λ sopstvena vrednost matrice A tada je $\lambda^\dagger$ sopstvena vrednost matrice A^D.*

Drazinov inverz može da se izračuna korišćenjem Jordanove normalne forme matrice A. Ovaj metod je opisan u sledećoj teoremi.

Teorema 3.1.27. *Posmatrajmo Jordanovu normalnu formu matrice $A \in \mathbb{C}^{n\times n}$,*

$$A = PJP^{-1} = P \begin{bmatrix} J_1 & \mathbb{O} \\ \mathbb{O} & J_0 \end{bmatrix} P^{-1}, \tag{3.11}$$

gde J_1 i J_0 sadrže Jordanove blokove koji odgovaraju sopstvenim vrednostima različitim od nule i jednakim nula, respektivno. Tada važi

$$A^D = P^{-1} \begin{bmatrix} J_1^{-1} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} P. \tag{3.12}$$

Napomenimo da važi i opštiji rezultat. Ako A možemo napisati u obliku

$$A = PJP^{-1} = P \begin{bmatrix} C & \mathbb{O} \\ \mathbb{O} & N \end{bmatrix} P^{-1},$$

gde je C regularna a N nilpotentna matrica (postoji broj n takav da je $A^n = \mathbb{O}$) tada je

$$A^D = P^{-1} \begin{bmatrix} C^{-1} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} P.$$

Jordanova dekompozicija (3.11) je specijalan slučaj dekompozicije (3.12).

Još neka svojstva Drazinovog inverza kao i definicije i svojstva ostalih spektralnih inverza mogu se naći na primer u [10, 108, 148].

3.2 Osnovni metodi za izračunavanje generalisanih inverza

U ovom odeljku daćemo neke poznate metode za računanje različitih generalisanih inverza matrica. Neki od ovih metoda (ili bar ideje na kojima su bazirani) su već objašnjeni u predhodnom poglavlju. Napomenimo da postoji jako veliki broj različitih metoda za izračunavanje različitih klasa generalisanih inverza. Kratak prikaz ovih metoda može se naći na primer u [10, 148].

3.2.1 Metodi bazirani na faktorizacijama potpunog ranga

Pokazaćemo nekoliko reprezentacija generalisanih inverza u kojima figuriše faktorizacija potpunog ranga određenih matrica [125].

Podsetimo se da je reprezentacija MP inverza preko faktorizacija potpunog ranga data u Teoremi 3.1.15 iz predhodnog poglavlja. Sličan rezultat za težinske MP inverze dat je u Lemi 3.1.24.

Sledeća lema uvodi dve opšte reprezentacije $\{1,2\}$ inverza, date u radovima [109, 108, 125].

Teorema 3.2.1. *Neka je $A = PQ$ faktorizacija potpunog ranga matrice $A \in \mathbb{C}_r^{m\times n}$. Neka su U i V matrice koje prolaze skupovima matrica formata $m \times m$ i $n \times n$, respektivno. Takođe, neka matrice W_1 i W_2 prolaze skupovima matrica formata $n \times r$ i $r \times m$, respektivno. Tada skupovi*

$$\begin{aligned} S_1 &= \{VQ^*(P^*UAVQ^*)^{-1}P^*U \mid \quad U \in \mathbb{C}^{m\times m},\ V \in \mathbb{C}^{n\times n},\ \mathrm{rang}(P^*UAVQ^*) = r\} \\ S_2 &= \{W_1(W_2AW_1)^{-1}W_2 \mid \quad W_1 \in \mathbb{C}^{n\times r},\ W_2 \in \mathbb{C}^{r\times n},\ \mathrm{rang}(W_2AW_1) = r\} \end{aligned}$$

zadovoljavaju $S_1 = S_2 = A\{1,2\}$.

Napomenimo da je u radu [109], Radić dao opšte rešenje sistema matričnih jednačina (1), (2) iz (3.2) u sledećem obliku

$$X = W_1(QW_1)^{-1}(W_2P)^{-1}W_2 = W_1(W_2AW_1)^{-1}W_2.$$

Ovde smo sa $A = PQ$ označili proizvoljnu faktorizaciju potpunog ranga dok matrice W_1 i W_2 prolaze skupovima matrica formata $n \times r$ i $r \times m$, respektivno i pritom zadovoljavaju uslov $\text{rang}(QW_1) = \text{rang}(W_2P) = \text{rang}(A)$.

Takodje, navešćemo sledeće opšte reprezentacije za $\{1,2,3\}$ i $\{1,2,4\}$ inverze.

Teorema 3.2.2. *Neka je $A = PQ$ faktorizacija potpunog ranga matrice A i matrice W_1 i W_2 zadovoljavaju uslove Teoreme 3.2.1. Tada važi:*

(1) *Opšte rešenje sistema matričnih jednačina (1), (2), (3) iz skupa jednačina* (3.2) *dato je pomoću* [**109**]
$$W_1(QW_1)^{-1}(P^*P)^{-1}P^* = W_1(P^*AW_1)^{-1}P^* \quad , \tag{3.13}$$

(2) *Opšte rešenje sistema matričnih jednačina (1), (2), (4) iz skupa jednačina* (3.2) *dato je pomoću* [**109**]
$$Q^*(QQ^*)^{-1}(W_2P)^{-1}W_2 = Q^*(W_2AQ^*)^{-1}W_2 \quad . \tag{3.14}$$

Opšta reprezentacija klase $\{2\}$ inverza uvedena je u radu Stanimirovića [125] i data u sledećoj teoremi.

Teorema 3.2.3. [125] (Stanimirović, 1998) *Skup svih $\{2\}$ inverza date matrice $A \in \mathbb{C}_r^{m\times n}$ odredjen je sledećom relacijom*

$$A\{2\} = \{W_1(W_2AW_1)^{-1}W_2,\ W_1 \in \mathbb{C}^{n\times t},\ W_2 \in \mathbb{C}^{t\times m},\ \text{rang}(W_2AW_1) = t,\ t = 1,\dots,r\}.$$

Na kraju ovog odeljka razmotrićemo faktorizacije potpunog ranga za Drazinov inverz. Faktorizacija potpunog ranga se u ovom slučaju računa za rang invarijantne matrice A^l, gde je $l \geq \text{ind}A$. Do ovog rezultata je, takodje, došao Stanimirović u radu [125].

Teorema 3.2.4. [125] (Stanimirović, 1998) *Neka je $A \in \mathbb{C}_r^{n\times n}$, $k = \text{ind}A$ i $l \geq k$ proizvoljan ceo broj. Ako je $A^l = PQ$ faktorizacija potpunog ranga matrice A^l, tada je Drazinov inverz matrice A jednak*
$$A^D = P(QAP)^{-1}Q. \tag{3.15}$$

Postoji mnogo metoda koji su bazirani na specijalnim faktorizacijama potpunog ranga, na primer LU faktorizaciji. Rezultati izloženi u ovom odeljku važe za proizvoljne faktorizacije potpunog ranga.

3.2.2 Blokovske reprezentacije generalisanih inverza

Sada ćemo razmotriti reprezentacije nekih klasa generalisanih inverza pomoću odgovarajućih blok dekompozicija (reprezentacija) matrice A.

Lema 3.2.5. *Za zadatu matricu $A \in \mathbb{C}_r^{m\times n}$ postoje regularne matrice R i G, permutacione matrice E i F i unitarne matrice U i V, odgovarajućih dimenzija, tako da važi* [162]

$$(T_1)\quad RAG = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1 \qquad (T_2)\quad RAG = \begin{bmatrix} B & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_2$$

$$(T_3)\quad RAF = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_3 \qquad (T_4)\quad EAG = \begin{bmatrix} I_r & \mathbb{O} \\ K & \mathbb{O} \end{bmatrix} = N_4$$

$$(T_5)\quad UAG = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1 \qquad (T_6)\quad RAV = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1$$

$$(T_7)\quad UAV = \begin{bmatrix} B & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_2 \qquad (T_8)\quad UAF = \begin{bmatrix} B & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_5$$

$$(T_9)\quad EAV = \begin{bmatrix} B & \mathbb{O} \\ K & \mathbb{O} \end{bmatrix} = N_6$$

$(T_{10a})\quad EAF = \begin{bmatrix} A_{11} & A_{11}T \\ SA_{11} & SA_{11}T \end{bmatrix}$, *gde su S i T multiplikatori dobijeni iz uslova* [93]

$$T = A_{11}^{-1}A_{12}, \qquad S = A_{21}A_{11}^{-1};$$

$$(T_{10b})\quad EAF = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{21}A_{11}^{-1}A_{12} \end{bmatrix}.\ [142]$$

(T_{11}) *Transformacija sličnosti kvadratnih matrica* [114]

$$RAR^{-1} = RAEE^*R^{-1} = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} E^*R^{-1} = \begin{bmatrix} T_1 & T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix}.$$

Iz blokovskih dekompozicija matrica sledi čitav niz reprezentacija za različite klase generalisanih inverza. Sledi prikaz poznatih blokovskih reprezentacija za različite klase generalisanih inverza, koje su razvijene u [119, 161].

Teorema 3.2.6. [161] (Zielke 1979) *Ako je $A \in \mathbb{C}_r^{m\times n}$ i ako su $R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$ i $G = \begin{bmatrix} G_1, & G_2 \end{bmatrix}$, matrice kojima se A transformiše u normalnu formu $A = R^{-1}\begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} G^{-1}$, tada je*

$$A^{(1)} = G\begin{bmatrix} I_r & X \\ Y & Z \end{bmatrix} R$$

$$A^{(1,2)} = G\begin{bmatrix} I_r & X \\ Y & YX \end{bmatrix} R = G\begin{bmatrix} I_r \\ Y \end{bmatrix}\begin{bmatrix} I_r, & X \end{bmatrix} R$$

$$A^{(1,3)} = G\begin{bmatrix} I_r & -R_1R_2^\dagger \\ Y & Z \end{bmatrix} R$$

$$A^{(1,4)} = G\begin{bmatrix} I_r & X \\ -G_2^\dagger G_1 & Z \end{bmatrix} R$$

$$A^{(1,2,3)} = G\begin{bmatrix} I_r & -R_1R_2^\dagger \\ Y & -YR_1R_2^\dagger \end{bmatrix} R = G\begin{bmatrix} I_r \\ Y \end{bmatrix}\begin{bmatrix} I_r, & -R_1R_2^\dagger \end{bmatrix} R$$

$$A^{(1,2,4)} = G\begin{bmatrix} I_r & X \\ -G_2^\dagger G_1 & -G_2^\dagger G_1 X \end{bmatrix} R = G\begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix}\begin{bmatrix} I_r, & X \end{bmatrix} R$$

$$A^{(1,3,4)} = G\begin{bmatrix} I_r & -R_1R_2^\dagger \\ -G_2^\dagger G_1 & Z \end{bmatrix} R$$

$$\begin{aligned} A^\dagger = G\begin{bmatrix} I_r & -R_1R_2^\dagger \\ -G_2^\dagger G_1 & G_2^\dagger G_1 R_1 R_2^\dagger \end{bmatrix} R &= G\begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix}\begin{bmatrix} I_r, & -R_1R_2^\dagger \end{bmatrix} R = \\ &= (G_1 - G_2G_2^\dagger G_1)(R_1 - R_1R_2^\dagger R_2). \end{aligned}$$

Teorema 3.2.7. [161] (Zielke 1979) *Ako je* $A \in \mathbb{C}_r^{m\times n}$ *i ako su* $R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$ *i* $G = \begin{bmatrix} G_1, & G_2 \end{bmatrix}$, *matrice kojima se* A *transformiše u normalnu formu* $A = R^{-1}\begin{bmatrix} B & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} G^{-1}$, *tada je:*

$$A^{(1)} = G\begin{bmatrix} B^{-1} & X \\ Y & Z \end{bmatrix} R$$

$$A^{(1,2)} = G\begin{bmatrix} B^{-1} & X \\ Y & YBX \end{bmatrix} R = G\begin{bmatrix} B^{-1} \\ Y \end{bmatrix}\begin{bmatrix} I_r, & BX \end{bmatrix} R$$

$$A^{(1,3)} = G\begin{bmatrix} B^{-1} & -B^{-1}R_1R_2^\dagger \\ Y & Z \end{bmatrix} R$$

$$A^{(1,4)} = G\begin{bmatrix} B^{-1} & X \\ -G_2^\dagger G_1 B^{-1} & Z \end{bmatrix} R$$

$$A^{(1,2,3)} = G\begin{bmatrix} B^{-1} & -B^{-1}R_1R_2^\dagger \\ Y & -YR_1R_2^\dagger \end{bmatrix} R = G\begin{bmatrix} B^{-1} \\ Y \end{bmatrix}\begin{bmatrix} I_r, & -R_1R_2^\dagger \end{bmatrix} R$$

$$A^{(1,2,4)} = G\begin{bmatrix} B^{-1} & X \\ -G_2^\dagger G_1 B^{-1} & -G_2^\dagger G_1 X \end{bmatrix} R = G\begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix}\begin{bmatrix} B^{-1}, & X \end{bmatrix} R$$

$$A^{(1,3,4)} = G\begin{bmatrix} B^{-1} & -B^{-1}R_1R_2^\dagger \\ -G_2^\dagger G_1 B^{-1} & Z \end{bmatrix} R$$

$$\begin{aligned} A^\dagger = G&\begin{bmatrix} B^{-1} & -B^{-1}R_1R_2^\dagger \\ -G_2^\dagger G_1 B^{-1} & G_2^\dagger G_1 B^{-1} R_1 R_2^\dagger \end{bmatrix} R = \\ &= G\begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix} B^{-1}\begin{bmatrix} I_r, & -R_1R_2^\dagger \end{bmatrix} R = \\ &= (G_1 - G_2G_2^\dagger G_1)B^{-1}(R_1 - R_1R_2^\dagger R_2). \end{aligned}$$

Napomenimo da se reprezentacije generalisanih inverza pomoću ostalih blok dekompozicija iz Leme 3.2.5 mogu naći u radovima [119, 161].

3.2.3 Metod Žukovskog

U [163] se polazi od poznatih rekurentnih jednačina za rešavanje linearnog sistema $Ax = y$, gde je $A \in \mathbb{C}_m^{m\times n}$.

$$\begin{aligned} x_{t+1} &= x_t + \frac{\gamma_t a_{t+1}^*}{a_{t+1}\gamma_t a_{t+1}^*}\left(y_{t+1} - a_{t+1}x_t\right), \quad x_0 = \mathbb{O}, \\ \gamma_{t+1} &= \gamma_t - \frac{\gamma_t a_{t+1}^* a_{t+1}\gamma_t}{a_{t+1}\gamma_t a_{t+1}^*}, \quad \gamma_0 = I_n. \end{aligned}$$

U istom radu je uvedena generalizacija ovih rekurentnih jednačina, čime je nadjeno rešenje saglasnog sistema linearnih algebarskih jednačina $Ax = y$, u slučaju matrice $A \in \mathbb{C}_r^{m\times n}$, $r \leq m \leq n$. Pokazano je da se x_t, γ_t, $t = 1, \ldots, m$ mogu definisati na sledeći način

$$\begin{aligned} x_{t+1} &= x_t + \gamma_t a_{t+1}^* \left(a_{t+1}\gamma_t a_{t+1}^*\right)^\dagger \left(y_{t+1} - a_{t+1}x_t\right), \quad x_0 = \mathbb{O}, \\ \gamma_{t+1} &= \gamma_t - \gamma_t a_{t+1}^* \left(a_{t+1}\gamma_t a_{t+1}^*\right)^\dagger a_{t+1}\gamma_t, \quad \gamma_0 = I_n, \end{aligned}$$

gde je

$$\left(a_{t+1}\gamma_t a_{t+1}^*\right)^\dagger = \begin{cases} \frac{1}{a_{t+1}\gamma_t a_{t+1}^*}, & a_{t+1}\gamma_t a_{t+1}^* > 0, \\ 0, & a_{t+1}\gamma_t a_{t+1}^* = 0. \end{cases}$$

Napomenimo da je $a_{t+1}\gamma_t a_{t+1}^* = 0$ ako i samo ako je vrsta a_{t+1} linearno zavisna od vrsta $a_1, \ldots, a_t$.

Odavde je razvijen metod za izračunavanje MP inverza matrice $A \in \mathbb{C}^{m\times n}$, $m \leq n$ [164].

Neka je $\Gamma_t,\ t = 1, \ldots, n$ niz $n \times n$ matrica, definisan pomoću

$$\Gamma_{t+1} = \Gamma_t - \Gamma_t b_{t+1} \left(b_{t+1}^* \Gamma_t b_{t+1}\right)^\dagger b_{t+1}^* \gamma_t, \quad \gamma_0 = I_n. \tag{3.16}$$

Ovde smo sa $b_t,\ t = 1, \ldots, n$ označili kolone matrice A, dok smo sa $c_t,\ t = 1, \ldots, n$ označili vrste matrice $I_n - \Gamma_n$. Posmatrajmo niz matrica X_t i γ_t dimenzija $n \times n$, definisanih na sledeći način

$$\begin{aligned} \gamma_{t+1} &= \gamma_t - \gamma_t a_{t+1}^* \left(a_{t+1}\gamma_t a_{t+1}^*\right)^\dagger a_{t+1}\gamma_t, \quad \gamma_0 = I_n, \\ X_{t+1} &= X_t + \gamma_t a_{t+1}^* \left(a_{t+1}\gamma_t a_{t+1}^*\right)^\dagger \left(c_{t+1} - a_{t+1}x_t\right), \quad X_0 = \mathbb{O}. \end{aligned} \tag{3.17}$$

Sledeća teorema povezuje predhodno definisani niz matrica X_t sa MP inverzom $A^\dagger$ matrice A i pokazuje ispravnost konstruisanog metoda za računanje MP inverza.

Teorema 3.2.8. [164] (Žukovski, Lipcer 1972) *Neka je niz X_t, $t = 0, \ldots, n$ definisan kao u* (3.16) *i* (3.17). *Tada je $X_n = A^\dagger$.*

3.3 Leverrier-Faddeev metod

U ovom poglavlju proučićemo poznati Leverrier-Faddev metod. Ovaj metod je reotkrivan i modifikovan više puta. U originalu ovaj metod računa koeficijente karakterističnog polinoma matrice A.

Godine 1840., francuski naučnik U.J.J. Leverrier [78] uspostavio je vezu između karakterističnog polinoma i Newton-Girardovih relacija. J.M. Souriau i J.S. Frame [36], su nezavisno jedan od drugog modifikovali Leverrierov metod u oblik koji on danas ima. Paul Horst 1935 zajedno sa Faddeevom i Sominskiim [34] 1949 su takođe reotkrili ovaj metod. Zato se on danas naziva Leverrier-Faddeev metod a takođe i Souriau-Frame metod. Gower [47, 48] je dalje razvijao ovaj algoritam i pokazao kako on može biti od koristi ne samo za računanje već i za izvođenje teorijskih rezultata. Iako je algoritam lep i elegantan, praktična primena mu je delom limitirana zbog loše uslovljenosti.

Napomenuli smo da se ovaj metod u originalu koristio za računanje koeficijenata karakterističnog polinoma matrice A. Kasnije je pokazano da ovaj metod može da se proširi na računanje raznih klasa generalisanih inverza (Moore-Penroseov inverz [27], Drazinov inverz [43], [57], kao i čitava klasa drugih generalisanih inverza [134]). Iz razloga kompletnosti, sve poznate rezultate ćemo pomenuti a pojedine i dokazati.

3.3.1 Karakteristični polinom

Podsetimo se da je za kvadratnu matricu $A \in \mathbb{C}^{n\times n}$, njen *karakteristični polinom* $P_A(x)$ jednak

$$P_A(x) = \det[xI_n - A].$$

Formulisaćemo dve poznate teoreme iz kojih ćemo izvesti Leverrier-Faddeev metod.

Teorema 3.3.1. (Newton-Girardove relacije) *Neka je $P(x)$ polinom stepena n, i neka je*

$$P(x) = a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n. \tag{3.18}$$

Označimo sa $x_1, x_2, \ldots, x_n$ korene polinoma $P(x)$ i neka je $\sigma_k = \sum_{i=1}^n x_i^k$ za svako $k = 1, 2, \ldots, n$. Tada je sledeća relacija

$$ka_k + \sigma_1 a_{k-1} + \ldots + \sigma_{k-1}a_1 + \sigma_k a_0 = 0, \tag{3.19}$$

zadovoljena za svako $k = 1, 2, \ldots, n$.

Neka je $A \in \mathbb{C}^{n\times n}$ proizvoljna kvadratna matrica. Posmatrajmo karakteristični polinom $P_A(x)$ u obliku (3.18). Na osnovu Teoreme 3.3.1 (relacije (3.19)) možemo zapisati koeficijent a_k na sledeći način

$$a_k = -\frac{1}{k}\left(\sigma_k + \sigma_{k-1}a_1 + \ldots + \sigma_1 a_{k-1}\right). \tag{3.20}$$

Imamo da je $\sigma_k = \sum_{i=1}^n \lambda_i^k = \operatorname{tr}(A^k)$ za $k = 1, 2, \ldots, n$, gde su $\lambda_1, \lambda_2, \ldots, \lambda_n$ sopstvene vrednosti matrice A. Zamenom u (3.20) dobijamo

$$a_k = -\frac{1}{k}\operatorname{tr}(A^k + a_1A^{k-1} + \ldots + a_{k-1}A) \tag{3.21}$$

Za svako $k = 1, 2, \ldots, n$ označimo

$$B_k = A^k + a_1A^{k-1} + \ldots + a_{k-1}A + a_kI_n.$$

i

$$A_k = AB_{k-1} \tag{3.22}$$

Tada relaciju (3.21) možemo zapisati u obliku

$$a_k = -\frac{1}{k} Tr(A_k). \tag{3.23}$$

Sledeća relacija trivijalno važi

$$B_k = B_{k-1} + a_k I_n. \tag{3.24}$$

Svi koeficijenti a_k, $k = 1, 2, \ldots, n$ karakterističnog polinoma $P_A(x)$ mogu da se izračunaju uzastopnom primenom relacija (3.22), (3.23) i (3.24). Prema tome možemo konstruisati Algoritam 3.3.1 i na osnovu predhodnog razmatranja imamo da važi sledeća teorema.

Teorema 3.3.2. *Za svaku matricu $A \in \mathbb{C}^{m\times n}$, izlaz Algoritma 3.3.1 je karakteristični polinom $P_A(x)$ matrice A.*

Algoritam 3.3.1 Leverrier-Faddeev metod za izračunavanje karakterističnog polinoma

Input: Proizvoljna kvadratna matrica $A \in \mathbb{C}^{n\times n}$.

$a_0 := 1$
$A_0 := \mathbb{O}$
$B_0 := I_n$
for $i := 1$ to n **do**
  $A_i := AB_{i-1}$
  $a_i := -\operatorname{tr}(A_i)/i$
  $B_i := A_i + a_i I_n$
end for
return $P_A(x) := a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1}x + a_n$

3.3.2 Moore-Penroseov inverz

Sada ćemo izvesti algoritam za računanje MP inverza date matrice A koji je baziran na Algoritmu 3.3.1. Koristićemo sledeću, dobro poznatu teoremu.

Teorema 3.3.3. (Cayley-Hamilton) *Za svaku matricu $A \in \mathbb{C}^{n\times n}$ važi $P_A(A) = \mathbb{O}$, tj. matrica A je koren svog karakterističnog polinoma.*

Decell je pokazao [27] da se MP inverz $A^\dagger$ može direktno dobiti iz karakterističnog polinoma $P_{AA^*}(x)$ matrice AA^*. Sledeća teorema upravo opisuje tu vezu. Iz razloga kompletnosti dajemo dokaz Decellovog rezultata koji je različit od onog datog u [27].

Teorema 3.3.4. [27] (Decell 1965) *Neka je $A \in \mathbb{C}^{m\times n}$ i*

$$P_{AA^*}(x) = \det[xI_n - AA^*] = a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1}x + a_n, \quad a_0 = 1,$$

Neka je k maksimalni indeks takav da je $a_k \neq 0$ (tj. da je $a_k \neq 0$ i $a_{k+1} = \ldots = a_n = 0$). Ako je $k > 0$ onda je MP inverz matrice A jednak

$$A^\dagger = -a_k^{-1} A^* [(AA^*)^{k-1} + a_1 (AA^*)^{k-2} + \ldots + a_{k-2} AA^* + a_{k-1}] \tag{3.25}$$

U suprotnom, ako je $k = 0$ onda važi $A^\dagger = \mathbb{O}$.

Dokaz. Ako je $k = 0$ onda je $A = \mathbb{O}$ pa je zbog toga i $A^\dagger = \mathbb{O}$. Pretpostavimo da je $k \neq 0$.

Direktna primena Cayley-Hamiltonove teoreme na matricu AA^* daje

$$(AA^*)^n + a_1 (AA^*)^{n-1} + \ldots + a_{k-1} (AA^*)^{n-k+1} + a_k (AA^*)^{n-k} = \mathbb{O}. \tag{3.26}$$

Posmatrajmo dekompoziciju matrice A (na osnovu Leme 3.1.3 takva dekompozicija postoji) u obliku $A = Q^* R' P$, gde su $Q \in \mathbb{C}^{n \times n}$ i $P \in \mathbb{C}^{n \times n}$ unitarne matrice i R' dato pomoću

$$R' = \begin{bmatrix} R'_{11} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix}, \quad R'_{11} \in \mathbb{C}_k^{k \times k}. \tag{3.27}$$

Tada važi $AA^* = Q^* R Q$, pri čemu $R = R' R'^*$ možemo predstaviti na isti način kao R' u (3.27)

$$R = \begin{bmatrix} R_{11} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix}, \quad R_{11} = R'_{11} R'^*_{11} \in \mathbb{C}_k^{k \times k}. \tag{3.28}$$

Pošto je $(AA^*)^i = Q^* R^i Q$ za svako $i = 1, 2, 3, \ldots$, zamenom u (3.26) dobijamo

$$Q^* \left[R^n + a_1 R^{n-1} + \ldots + a_{k-1} R^{n-k+1} + a_k R^{n-k} \right] Q = \mathbb{O}.$$

Iz činjenice da je Q regularna matrica i da je R oblika (3.28), važi

$$R_{11}^n + a_1 R_{11}^{n-1} + \ldots + a_{k-1} R_{11}^{n-k+1} + a_k R_{11}^{n-k} = \mathbb{O}.$$

Množenjem sa $R_{11}^{-(n-k)}$ obe strane predhodne jednačine dobijamo

$$R_{11}^k + a_1 R_{11}^{k-1} + \ldots + a_{k-1} R_{11} + a_k I_k = \mathbb{O}. \tag{3.29}$$

Označimo

$$I_{m,k} = \begin{bmatrix} I_k & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} \in \mathbb{C}^{m \times m}.$$

Jednačina (3.29) važi i kada se R_{11} zameni sa R i I_k sa $I_{m,k}$.

$$R^k + a_1 R^{k-1} + \ldots + a_{k-1} R + a_k I_{m,k} = \mathbb{O}. \tag{3.30}$$

Množenjem jednačine (3.30) sa Q^* sa leve strane i sa Q sa desne strane dobijamo

$$(AA^*)^k + a_1 (AA^*)^{k-1} + \ldots + a_{k-1} (AA^*) + a_k Q^* I_{m,k} Q = \mathbb{O}. \tag{3.31}$$

Poslednja relacija je ekvivalentna sa

$$-a_k^{-1} AA^* \left[(AA^*)^{k-1} + a_1 (AA^*)^{k-2} + \ldots + a_{k-2} (AA^*) + a_{k-1} \right] = Q^* I_m Q. \tag{3.32}$$

Primenom Leme 3.1.13 sada dobijamo

$$A^\dagger = P^* R^\dagger Q, \quad R^\dagger = \begin{bmatrix} R_{11}^{-1} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix}, \tag{3.33}$$

Množenjem obe strane jednačine (3.32) sa $A^\dagger$ i korišćenjem $A^\dagger A A^* = A^*$ (Lema 3.1.12) dokazujemo tvrđenje teoreme. $\square$

Sada se iz Teoreme 3.3.4 i Algoritma 3.3.1, lako konstruiše algoritam za izračunavanje MP inverza (Algoritam 3.3.2).

Algoritam 3.3.2 Leverrier-Faddeev metod za računanje MP inverza

Input: Matrica $A \in \mathbb{C}^{m\times n}$.

1: $a_0 := 1$
2: $A_0 := \mathbb{O}$
3: $B_0 := I_n$
4: **for** $i := 1$ to n **do**
5: $\quad A_i := AA^* B_{i-1}$
6: $\quad a_i := -\operatorname{tr}(A_i)/i$
7: $\quad B_i := A_i + a_i I_n$
8: **end for**
9: $k := \max\{i \mid a_i \neq 0, i = 0, \dots, n\}$
10: **if** $k = 0$ **then**
11: $\quad$ **return** $A^\dagger := \mathbb{O}$
12: **else**
13: $\quad$ **return** $A^\dagger := a_k^{-1} A^* B_{k-1}$
14: **end if**

Sada ćemo izvršiti analizu složenosti Algoritma 3.3.2. Telo petlje date u koracima 4-8 se ponavlja n puta. Ono se sastoji od dva množenja matrica, jednog izračunavanja traga i jednog izračunavanja zbira dve matrice. Prema tome, složenost tela ove petlje je $\mathcal{O}(n^3)$ (ukoliko pretpostavimo da se množenje matrica odvija u vremenu $\mathcal{O}(n^3)$). Prema tome, složenost petlje u koracima 4-8 je $\mathcal{O}(n^3 \cdot n) = \mathcal{O}(n^4)$.

Složenost koraka 9 je $\mathcal{O}(n)$ dok je složenost koraka 13 jednaka $\mathcal{O}(n^3)$. Prema tome ukupna složenost Algoritma 3.3.2 je $\mathcal{O}(n^4)$.

3.3.3 Drazinov inverz

Slično kao i u slučaju MP inverza, Drazinov inverz A^D može se izračunati korišćenjem karakterističnog polinoma $P_A(x)$ matrice A. U radovima [43, 57] je data sledeća reprezentacija Drazinovog inverza.

Teorema 3.3.5. [43], [57] (Greville 1973). *Neka je data kvadratna matrica $A \in \mathbb{C}^{n\times n}$. Pretpostavimo da je*

$$P_A(x) = \det[xI_n - A] = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1}x + a_n, \quad a_0 = 1,$$

Razmotrimo sledeći niz matrica formata $n \times n$ definisanih koeficijentima a_i i stepenima matrice A

$$B_j = a_0 A^j + a_1 A^{j-1} + \cdots + a_{j-1} A + a_j I_n, \quad a_0 = 1, \quad j = 0, \dots, n \tag{3.34}$$

Nek je r minimalni ceo broj takav da je $B_r = \mathbb{O}$, neka je t najveći ceo broj takav da je $a_t \neq 0$ i neka je $k = r - t$. Tada Drazinov inverz A^D matrice A možemo predstaviti na sledeći način

$$A^D = (-1)^{k+1} a_t^{-k-1} A^k B_{t-1}^{k+1}. \tag{3.35}$$

Sada iz Teoreme 3.3.5 direktno dobijamo sledeće uopštenje Leverrier-Faddevog metoda za računanje Drazinovog inverza (Algoritam 3.3.3).

Algoritam 3.3.3 Leverrier-Faddeev method za računanje Drazinovog inverza

Input: Matrica $A \in \mathbb{C}^{n \times n}$.

1: $a_0 := 1$
2: $A_0 := \mathbb{O}$
3: $B_0 := I_n$
4: **for** $i := 1$ to n **do**
5: $\quad A_i := AB_{i-1}$
6: $\quad a_i := -\operatorname{tr}(A_i)/i$
7: $\quad B_i := A_i + a_i I_n$
8: **end for**
9: $k := \max\{i \mid a_i \neq 0, i = 0, \dots, n\}$
10: $t := \min\{i \mid B_i = \mathbb{O}, i = 0, \dots, n\}$
11: $r := t - k$
12: **return** $A^D := (-1)^{r+1} a_k^{-r-1} A^r B_{k-1}^{r+1}$

Sprovešćemo analizu složenosti Algoritma 3.3.3. Slično kao i u slučaju Algoritma 3.3.2, petlja u koracima 4-8 ima složenost $\mathcal{O}(n^4)$. Složenost koraka 10 je $\mathcal{O}(n \cdot n^2)$. U koraku 12 potrebno je izračunati r-ti i $r+1$-vi stepen matrica A i B_{k-1} respektivno. Računanje ovih stepena moguće je u vremenu $\mathcal{O}(n^3 \log n)$ zato što je $r \leq n$.

To znači da je ukupna složenost Algoritma 3.3.3 jednaka je $\mathcal{O}(n^4)$. Prema tome, dobili smo istu složenost kao i u slučaju Algoritma 3.3.2 za računanje MP inverza.

3.3.4 Ostali generalisani inverzi

U predhodna dva odeljka pokazali smo dva uopštenja Leverrier-Faddevog metoda za računanje MP i Drazinovog inverza date matrice. Ova ideja može se dalje generalisati do metoda Leverrier-Faddevog tipa za računanje široke klase generalisanih inverza.

Na taj način dobijamo Algoritam 3.3.4. Nasuprot Algoritmu 3.3.2 kao i Algoritmu 3.3.3 koji na ulazu uzimaju samo jednu matricu, Algoritam 3.3.4 uzima dve matrice R i T i jedan prirodni broj $e \in \mathbb{N}$. Ovaj metod je preuzet iz rada [134].

Sledeća teorema (preuzeta iz rada [134]) pokazuje šta je izlaz Algoritma 3.3.4 za različite vrednosti matrica R i T kao i broja e na ulazu.

Algoritam 3.3.4 Metod Leverrier-Faddeevog tipa za računanje široke klase generalisanih inverza

Input: Matrice $R, T \in \mathbb{C}^{n\times m}$ i pozitivan ceo broj $e \in \mathbb{N}$.

$a_0 := 1$
$A_0 := \mathbb{O}$
$B_0 := I_n$
for $i := 1$ to n **do**
  $A_i := TR^* B_{i-1}$
  $a_i := -\operatorname{tr}(A_i)/i$
  $B_i := A_i + a_i I_n$
end for
$k := \max\{i \mid a_i \neq 0, i = 0, \dots, n\}$
if $k = 0$ **then**
  return $X_e := \mathbb{O}$
else
  return $X_e := (-1)^e a_k^{-e} R^* B_{k-1}^e$
end if

Teorema 3.3.6. [134] **(Stanimirović 2003)** *Neka je A matrica formata $n \times m$ i neka je $A = PQ$ faktorizacija potpunog ranga ove matrice. Tada važi*

(1) *Ako je $R = T = A$ onda je $X_1 = A^\dagger$.*

(2) *Ako je $m = n$, $R^* = A^l$, $T = A$ i $l \geq \operatorname{ind}A$ onda važi $X_1 = A^D$.*

(3) *Ako je $T = A$ i $n > m = \operatorname{rank}A$ za proizvoljnu matricu R takvu da je AR^* regularna, tada važi $X_1 = A_R^{-1}$.*

(4) *Ako je $m = n$, $R^* = A^k$ i $T = I_n$ tada X_1 postoji ako i samo ako je $\operatorname{ind}A = k$ i $X_1 = AA^D$.*

(5) *U slučaju $m = n$, $e = l + 1$, $TR^* = A$, $R^* = A^l$ i $l \geq \operatorname{ind}A$ važi $X_e = A^D$.*

(6) *Za $m = n$, $e = 1$, $T = R^* = A^l$ i $l \geq \operatorname{ind}A$ imamo da je $X_1 = (A^D)^l$.*

(7) *$X_1 \in A\{2\}$ ako i samo ako $T = A, R = GH, G \in \mathbb{C}^{n\times t}, H \in \mathbb{C}^{t\times m}, \operatorname{rank}HAG = t$.*

(8) *$X_1 \in A\{1,2\}$ ako i samo ako $T = A, R = GH, G \in \mathbb{C}^{n\times r}, H \in \mathbb{C}^{r\times m}, \operatorname{rank}HAG = r = \operatorname{rank}A$.*

(9) *$X_1 \in A\{1,2,3\}$ ako i samo ako $T = A, R = GP^*, G \in \mathbb{C}^{n\times r}, \operatorname{rank}P^*AG = r = \operatorname{rank}A$.*

(10) *$X_1 \in A\{1,2,4\}$ ako i samo ako $T = A, R = Q^*H, H \in \mathbb{C}^{r\times n}, \operatorname{rank}HAQ^* = r = \operatorname{rank}A$.*

(11) *Ako je $T = A$ i $m > n = \operatorname{rank}A$ za proizvoljnu matricu R takvu da je R^*A regularna, onda važi $X_1 = A_L^{-1}$.*

Ako uradimo analizu složenosti Algoritma 3.3.4 na sličan način kao u slučaju Algoritma 3.3.2 i Algoritma 3.3.3 dobijamo da je ukupna složenost Algoritma 3.3.4 jednaka $\mathcal{O}(n^4 + n^3 \log e)$. Ovaj rezultat važi pod uslovom da je složenost množenja matrica jednaka $\mathcal{O}(n^3)$. Napomenimo samo da je složenost koraka 13 jednaka $\mathcal{O}(n^3 \log e)$. Ukoliko je $e \ll 10^n$, onda je ukupna složenost Algoritma 3.3.4 za ulaznu matricu $A \in \mathbb{C}^{n\times m}$ jednaka ponovo $\mathcal{O}(n^4)$.

3.4 Metod pregradjivanja

Ovo poglavlje je posvećeno još jednoj klasi dobro poznatih metoda za računanje generalisanih inverza, metodima pregradjivanja. Svi ovi metodi su konačni iterativni metodi. Zajedničko za ove metode je da se u k-toj iteraciji $k = 1, 2, \ldots, n$ računaju generalisani inverzi A_k^-, gde je A_k podmatrica matrice A koja se sastoji od prvih k kolona ove matrice. Zato ćemo matricu A_k pregraditi na sledeći način

$$A_k = [A_{k-1}\ a_k]. \tag{3.36}$$

gde je sa a_k označena k-ta kolona matrice A. Zato se ovi metodi nazvaju *metodi pregradjivanja.*

3.4.1 Moore-Penroseov inverz

Prvi metod u ovoj klasi metoda je Grevilleov metod pregradjivanja [44] za računanje MP inverza. Glavna ideja kod ovog metoda je da se $A_k^\dagger$ prikaže u funkciji od $A_{k-1}^\dagger$, A_{k-1} i a_k. Za $k = 2, \ldots, n$, neka su vektori d_k i c_k definisani na sledeći način

$$\begin{aligned} d_k &= A_{k-1}^\dagger a_k \\ c_k &= a_k - A_{k-1} d_k = a_k - A_{k-1} A_{k-1}^\dagger a_k \end{aligned} \tag{3.37}$$

Potrebna veza izmedju $A_k^\dagger$ sa jedne, i $A_{k-1}^\dagger$, A_{k-1} i a_k sa druge strane, data je sledećom teoremom.

Teorema 3.4.1. [10, 44] (Greville 1960) *Neka je $A \in \mathbb{C}^{m\times n}$. Korišćenjem predhodno definisane notacije, MP inverz matrice A_k $(k = 2, 3, \ldots, n)$ možemo izraziti na sledeći način*

$$A_k^\dagger = \begin{bmatrix} A_{k-1} & a_k \end{bmatrix}^\dagger = \begin{bmatrix} A_{k-1}^\dagger - d_k b_k^* \\ b_k^* \end{bmatrix}, \tag{3.38}$$

gde je

$$b_k^* = \begin{cases} c_k^\dagger, & c_k \neq \mathbb{O} \\ (1 + d_k^* d_k)^{-1} d_k^* A_{k-1}^\dagger, & c_k = \mathbb{O} \end{cases} \tag{3.39}$$

Možemo konstruisati Algoritam 3.4.1 za izračunavanje MP inverza, baziran na Teoremi 3.4.1. Startna vrednost za ovaj metod je

$$a_1^\dagger = A_1^\dagger = \begin{cases} (a_1^* a_1)^{-1} a_1^*, & a_1 \neq \mathbb{O} \\ \mathbb{O}, & a_1 = \mathbb{O} \end{cases}. \tag{3.40}$$

Algoritam 3.4.1 Metod pregradjivanja za računanje MP inverza

Input: Matrica $A \in \mathbb{C}^{m\times n}$.

1: $A_1^\dagger := a_1^\dagger := \begin{cases} (a_1^* a_1)^{-1} a_1^*, & a_1 \neq \mathbb{O} \\ \mathbb{O}, & a_1 = \mathbb{O} \end{cases}$
2: **for** $k := 2$ to n **do**
3: $\quad d_k := A_{k-1}^\dagger a_k$
4: $\quad c_k := a_k - A_{k-1} d_k$
5: $\quad$**if** $c_k = \mathbb{O}$ **then**
6: $\quad\quad b_k^* := (c_k^* c_k)^{-1} c_k^*$
7: $\quad$**else**
8: $\quad\quad b_k^* := (1 + d_k^* d_k)^{-1} d_k^* A_{k-1}^\dagger$
9: $\quad$**end if**
10: $\quad A_k^\dagger := \begin{bmatrix} A_{k-1}^\dagger - d_k b_k^* \\ b_k^* \end{bmatrix},$
11: **end for**
12: **return** $A^\dagger := A_n^\dagger$

Kao i za predhodno uvedene metode i ovde ćemo uraditi analizu složenosti Algoritma 3.4.1. Posmatrajmo petlju u koracima 2-11. Vremenski najzahtevnija operacija u telu ove petlje je množenje matrice i vektora u koracima 3, 4 i 8, kao i računanje proizvoda dva vektora u koraku 10. Složenost svih ovih operacija je $\mathcal{O}(m \cdot k)$. Prema tome, ukupna složenost Algoritma 3.4.1 je jednaka

$$\mathcal{O}\left(\sum_{k=2}^{n} m \cdot k\right) = \mathcal{O}(m \cdot n^2).$$

Napomenimo da postoje i druge varijante Algoritma 3.4.1. Na primer, Clineov metod [19, 148] računa MP inverz pregradjene matrice $A = [U \ V]$. Takodje, Nobleov metod [86, 88, 148] uopštava, kako Grevilleov, tako i Clineov metod i pod odredjenim uslovima daje MP ili Drazinov inverz blok matrice

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \tag{3.41}$$

3.4.2 $\{1\}$ inverz

Algoritam 3.4.1 možemo iskoristiti i za računanje $\{1\}$ inverza, pošto je $A^\dagger \in A\{1\}$. Medjutim, ukoliko nam je potrebno da izračunamo samo $\{1\}$ inverz matrice A, Algoritam 3.4.1 se može značajno uprostiti. Sledeća teorema daje metod pregradjivanja za računanje $\{1\}$ inverza date matrice A.

Teorema 3.4.2. [**10**] *Neka je $A \in \mathbb{C}^{m\times n}$ proizvoljna matrica. Tada za svako $k = 2, \ldots, n$ važi*

$$A_k^{(1)} = \begin{bmatrix} A_{k-1}^{(1)} - d_k b_k^* \\ b_k^* \end{bmatrix},$$

gde su d_k i b_k^* definisani na sledeći način

$$\begin{aligned}
d_k &= A_{k-1}^{(1)} a_k \\
c_k &= a_k - A_{k-1} d_k \\
b_k^* &= \begin{cases} \mathbb{O}, & c_k = \mathbb{O} \\ c_k^\dagger (I_m - A_{k-1} A_{k-1}^\dagger), & c_k \neq \mathbb{O} \end{cases} .
\end{aligned}$$

Složenost algoritma baziranog na Teoremi 3.4.2 može biti na sličan način izvedena kao u slučaju Algoritma 3.4.1 i takođe je jednaka $\mathcal{O}(n^3)$.

3.4.3 Težinski Moore-Penroseov inverz

Kao u slučaju Grevilleovog metoda pregradjivanja, sličan metod može se konstruisati za računanje težinskog MP inverza pregradjenih matrica. Ovaj metod su konstruisali Miao [87] i Wang i Chen [149, 147].

Posmatraćemo Hermitske, pozitivno definitne matrice $M \in \mathbb{C}^{m\times m}$ i $N \in \mathbb{C}^{n\times n}$. Glavna podmatrica $N_i \in \mathbb{C}^{i\times i}$ matrice N je pregradjena na sledeći način

$$N_i = \begin{bmatrix} N_{i-1} & l_i \\ l_i^* & n_{ii} \end{bmatrix}, \ i = 2, \dots, n, \tag{3.42}$$

gde je $l_i \in \mathbb{C}^{(i-1)\times 1}$ i n_{ii} broj. Neka je $N_1 = [n_{11}]$.

Da bi uprostili notaciju, označimo sa X_i težinski MP inverz $X_i = (A_i)_{MN_i}^\dagger$, za svako $i = 2, \dots, n$. Slično je $X_1 = (a_1^\dagger)_{M,N_1}$.

Teorema 3.4.3. **[149] (Wang, Chen 1989)** *Neka je $A \in \mathbb{C}^{m\times n}$, i neka su $M \in \mathbb{C}^{m\times m}$ i $N \in \mathbb{C}^{n\times n}$ pozitivno definitne Hermitske matrice. Tada za svako $i = 2, 3, \dots, n$, matrica X_i se može izračunati na sledeći način*

$$X_1 = \begin{cases} (a_1^* M a_1)^{-1} a_1^* M, & a_1 \neq 0, \\ a_1^*, & a_1 = 0, \end{cases} \tag{3.43}$$

$$X_i = \begin{bmatrix} X_{i-1} - (d_i + (I_n - X_{i-1} A_{i-1}) N_{i-1}^{-1} l_i) b_i^* \\ b_i^* \end{bmatrix}, \tag{3.44}$$

gde su vektori d_i, c_i i b_i^* definisani pomoću

$$d_i = X_{i-1} a_i \tag{3.45}$$

$$c_i = a_i - A_{i-1} d_i = (I - A_{i-1} X_{i-1}) a_i \tag{3.46}$$

$$b_i^* = \begin{cases} (c_i^* M c_i)^{-1} c_i^* M, & c_i \neq 0 \\ \delta_i^{-1} (d_i^* N_{i-1} - l_i^*) X_{i-1}, & c_i = 0, \end{cases} \tag{3.47}$$

dok je δ_i dato izrazom

$$\delta_i = n_{ii} + d_i^* N_{i-1} d_i - (d_i^* l_i + l_i^* d_i) - l_i^* (I - X_{i-1} A_{i-1}) N_{i-1}^{-1} l_i. \tag{3.48}$$

Takodje su u radu [149] autori iskoristili blok reprezentaciju matrice N_i za računanje njenog običnog inverza. Ovaj metod je baziran na sledećoj lemi.

Lema 3.4.4. [**149**] **(Wang, Chen 1989)** *Neka je N_i pregradjena kao u* (3.42)*. Pretpostavimo da su N_i i N_{i-1} regularne matrice. Tada važi*

$$N_i^{-1} = \begin{cases} \begin{bmatrix} N_{i-1} & l_i \\ l_i^* & n_{ii} \end{bmatrix}^{-1} = \begin{bmatrix} E_{i-1} & f_i \\ f_i^* & h_{ii} \end{bmatrix}, & i = 2, \dots, n, \\ \\ n_{11}^{-1}, & i = 1, \end{cases} \tag{3.49}$$

gde je

$$h_{ii} = \left(n_{ii} - l_i^* N_{i-1}^{-1} l_i\right)^{-1}, \tag{3.50}$$

$$f_i = -h_{ii} N_{i-1}^{-1} l_i, \tag{3.51}$$

$$E_{i-1} = N_{i-1}^{-1} + h_{ii}^{-1} f_i f_i^*. \tag{3.52}$$

Na osnovu Teoreme 3.4.3 i Leme 3.4.4, respektivno, možemo formulisati sledeće algoritme za računanje težinskog MP inverza matrice A i običnog inverza matrice $N_i^{-1} \in \mathbb{C}^{i \times i}$.

Algoritam 3.4.2 Metod pregradjivanja za računanje težinskog MP inverza

Input: Matrica $A \in \mathbb{C}^{m \times n}$ i Hermitske, pozitivno definitne matrice $M \in \mathbb{C}^{m \times m}$ i $N \in \mathbb{C}^{n \times n}$.

```
Izračunati X_1 = a_1^† na osnovu (3.43)
for i = 2 to n do
    Izračunati d_i korišćenjem (3.45)
    Izračunati c_i korišćenjem (3.46)
    Izračunati b_i^* korišćenjem (3.47) i (3.48)
    Primenom (3.44) izračunati X_i
end for
return X_n
```

Algoritam 3.4.3 Metod pregradjivanja za računanje inverza Hermitske, pozitivno definitne matrice

Input: Hermitska, pozitivno definitna matrica $N \in \mathbb{C}^{n \times n}$

```
N_1^{-1} := n_11^{-1}
for i := 2 to n do
    Izračunati h_ii korišćenjem (3.50)
    Izračunati f_i korišćenjem (3.51)
    Izračunati E_{i-1} korišćenjem (3.52)
    Izračunati N_i^{-1} korišćenjem (3.49)
end for
return N_n^{-1}
```

3.5 Izračunavanje generalisanih inverza u vremenu množenja matrica

U ovom poglavlju ćemo detaljno opisati nov metod za računanje MP kao i nekih $\{i, j, \ldots, k\}$ inverza, koji je baziran na generalisanoj Cholesky faktorizaciji [24]. Formulisaćemo rekurzivni algoritam za računanje generalisane Cholesky faktorizacije date simetrične, pozitivno semidefinitne matrice $A \in \mathbb{C}^{n\times n}$. Ovaj algoritam radi u vremenu množenja dve matrice istih dimenzija. Korišćenjem Strassenovog metoda za množenje i inverziju matrica zajedno sa opisanim metodom za računanje generalisane Cholesky faktorizacije dobijamo algoritme za računanje MP, $\{1,2,3\}$, $\{1,2,4\}$, $\{2,3\}$ i $\{2,4\}$ inverza. Vremenska složenost ovih algoritama je manja od $\mathcal{O}(n^3)$ i ujedno oni predstavljaju **vremenski najefikasnije algoritme za računanje generalisanih inverza**.

Rezultati izloženi u ovom poglavlju su originalni i preuzeti iz našeg još uvek neobjavljenog rada [105].

3.5.1 Strassenov metod za množenje i inverziju matrica

Neka su $A, B \in \mathbb{C}^{n\times n}$ proizvoljne matrice. Broj operacija potrebnih za izračunavanje proizvoda matrica A i B, $C = AB$ korišćenjem dobro poznatih formula

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj},$$

jednak je $2n^3 - n^2 = \mathcal{O}(n^3)$ (n^3 množenja i $n^3 - n^2$ sabiranja). U radu [138], V. Strassen je uveo metod za množenje dve matrice formata $n \times n$ čija je složenost jednaka $\mathcal{O}(n^{\log_2 7}) \approx \mathcal{O}(n^{2.807})$. Strassenov metod je rekurzivan i baziran na sledećoj propoziciji.

Propozicija 3.5.1. [**23**, **138**] **(Strassen 1969)** *Neka su date matrice $A, B \in \mathbb{C}^{2n\times 2n}$, neka je $C = AB$ i neka su matrice A, B i C razbijene na blokove na sledeći način*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},$$

gde su svi blokovi dimenzija $n \times n$. Uvedimo sledeće oznake

$$\begin{array}{ll} 1.\quad Q_1 = (A_{11} + A_{22})(B_{11} + B_{22}) & 5.\quad Q_5 = (A_{11} + A_{12})B_{22} \\ 2.\quad Q_2 = (A_{21} + A_{22})B_{11} & 6.\quad Q_6 = (A_{21} - A_{11})(B_{11} + B_{12}) \\ 3.\quad Q_3 = A_{11}(B_{12} - B_{22}) & 7.\quad Q_7 = (A_{12} - A_{22})(B_{21} + B_{22}) \\ 4.\quad Q_4 = A_{22}(B_{21} - B_{11}) & \end{array} \tag{3.53}$$

Tada blokove matrice C možemo izraziti na sledeći način

$$\begin{array}{ll} C_{11} = Q_1 + Q_4 - Q_5 + Q_7 & C_{12} = Q_3 + Q_5 \\ C_{21} = Q_2 + Q_4 & C_{22} = Q_1 + Q_3 - Q_2 + Q_6 \end{array} \tag{3.54}$$

Korišćenjem formula datih u Propoziciji 3.5.1, potrebno je izvršiti množenje 7 matrica formata $n \times n$ da bi dobili proizvod dve $2n \times 2n$ matrice. Ovih 7 množenja možemo izvršiti rekurzivno. Ostale operacije koje pritom moramo izvršiti imaju složenost $\mathcal{O}(n^2)$. Algoritam 3.5.1 realizuje ovaj metod.

Označimo sa $\mathrm{inv}(n)$ složenost Algoritma 3.5.2. Takodje, označimo sa $\mathrm{add}(n)$ složenost sabiranja dve $n \times n$ matrice, a sa $\mathrm{mul}(m, n, k)$ složenost množenja matrice formata $m \times n$ sa matricom formata $n \times k$.

Složenost Algoritma 3.5.1 možemo odrediti korišćenjem poznate Master Teoreme. Formulacija i dokaz Master Teoreme dati su npr. u [23]. Pošto je potrebno primeniti isti metod 7 puta na matrice čija je dimenzija upola manja, imamo da važi

$$\mathrm{mul}(n, n, n) = 7 \cdot \mathrm{mul}(n/2, n/2, n/2) + \mathcal{O}(n^2). \tag{3.55}$$

Primenom Master Teoreme imamo da je rešenje rekurentne jednačine (3.55) dato sa $\mathrm{mul}(n, n, n) = \mathcal{O}(n^{\log_2 7})$. U opštem slučaju važi $\mathrm{mul}(m, n, k) = \mathcal{O}((\max\{m, n, k\})^{\log_2 7})$.

Algoritam 3.5.1 Strassenov algoritam za množenje matrica

Input: Matrice $A, B \in \mathbb{C}^{n \times n}$.

if $n = 1$ **then**
  return $C = [c_{11}] := [a_{11} \cdot b_{11}]$
end if
if n je neparano **then**
  Dopuniti matrice A i B dodavanjem jedne nula vrste i jedne nula kolone.
  $n_{old} := n$
  $n := n + 1$
end if
Formirati blok dekompoziciju $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, $B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ gde su blokovi dimenzija $n/2 \times n/2$.
Izračunati vrednosti $Q_1, \dots, Q_7$ korišćenjem relacija (3.53) gde se množenje matrica vrši rekurzivno.
Izračunati vrednosti C_{11}, C_{12}, C_{21} i C_{22} korišćenjem relacija (3.54).
$C := \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$
if n_{old} je neparno **then**
  Izbaciti poslednju nula vrstu i nula kolonu matrice C
end if
return C

Postoje i drugi algoritmi za računanje proizvoda matrica $C = AB$ u vremenu ispod $\mathcal{O}(n^3)$. Trenutno najbolji algoritam je formulisan od strane Coppersmitha i Winograda u radu [22] i radi u vremenu $\mathcal{O}(n^{2.376})$.

Strassen je, takodje, uveo algoritam za inverziju date $n \times n$ matrice A koji ima istu složenost kao odgovarajući algoritam za množenje matrica. Ovaj algoritam je takođe baziran na blok dekompoziciji matrice A kao i sledećoj lemi.

Lema 3.5.2. [138] **(Strassen 1969)** *Ako je A data $n \times n$ matrica razložena na sledeći način*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad A_{11} \in \mathbb{C}^{k \times k} \tag{3.56}$$

i važi da su A i A_{11} regularne matrice, tada inverznu matricu $X = A^{-1}$ možemo izračunati na sledeći način [4]

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1} A_{12} S^{-1} A_{21} A_{11}^{-1} & -A_{11}^{-1} A_{12} S^{-1} \\ -S^{-1} A_{21} A_{11}^{-1} & S^{-1} \end{bmatrix}. \tag{3.57}$$

Matrice X_{11}, X_{12}, X_{21} i X_{22} u predhodnom izrazu možemo izračunati pomoću sledećih relacija

$$\begin{array}{llll} 1. & R_1 = A_{11}^{-1} & 7. & X_{12} = R_3 R_6 \\ 2. & R_2 = A_{21} R_1 & 8. & X_{21} = R_6 R_2 \\ 3. & R_3 = R_1 A_{12} & 9. & R_7 = R_3 X_{21} \\ 4. & R_4 = A_{21} R_3 & 10. & X_{11} = R_1 - R_7 \\ 5. & R_5 = R_4 - A_{22} & 11. & X_{22} = -R_6, \\ 6. & R_6 = R_5^{-1} & & \end{array} \tag{3.58}$$

primenjujući pritom minimalan broj operacija množenja matrica.

Primetimo da je matrica R_5 u relacijama (3.58) jednaka negativnoj vrednosti Schurovog komplementa matrice A_{11} u matrici A

$$R_5 = -(A_{22} - A_{21} A_{11}^{-1} A_{12}) = -S = -(A/A_{11}).$$

Formule (3.57) i (3.58) su primenljive ako i samo ako su A_{11} i Schurov komplement $S = (A/A_{11})$ regularne matrice. Formule (3.58) mogu biti iskorišćene za rekurzivno izračunavanje inverza A^{-1}. Relacije 1. i 6. u (3.58) sadrže inverziju matrica manjih dimenzija (tj. $k \times k$, odnosno, $(n-k) \times (n-k)$, respektivno). Primenjujući iste formule na podmatrice dobijamo rekurzivni metod za računanje inverzne matrice date matrice A. U tom slučaju rekurzija se nastavlja sve dok se ne dobije 1×1 matrica. Algoritam 3.5.2 realizuje ovaj metod.

Napomena 3.5.1. Ukoliko koristimo bilo koji algoritam za množenje dve matrice koji radi u vremenu $\mathcal{O}(n^{2+\epsilon})$, onda Algoritam 3.5.2, takodje radi, u vremenu $\mathcal{O}(n^{2+\epsilon})$, $0 < \epsilon < 1$ (ponovo na osnovu Master Teoreme). Specijalno, ukoliko se koristi Strassenov metod za množenje matrica (Algoritam 3.5.1), Algoritam 3.5.2 zahteva ukupno

$$\frac{6}{5} n^{\log_2 7} - \frac{1}{5} n \approx n^{2.807}$$

aritmetičkih operacija [3, 55, 138]. U suprotnom, ako koristimo uobičajeni algoritam za množenje matrica koji radi u vremenu $\mathcal{O}(n^3)$, onda je složenost Algoritma 3.5.2 jednaka $\mathcal{O}(n^3)$.

Algoritam 3.5.2 Strassenov metod za inverziju matrica

Input: Matrica $A \in \mathbb{C}^{n\times n}$, takva da su svi njeni glavni dijagonalni minori $A_{(S)}$ regularni.

1: **if** $n = 1$ **then**
2: **return** $X := [a_{11}^{-1}]$
3: **end if**
4: $k := \lfloor n/2 \rfloor$
5: Formirati blok dekompoziciju $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ tako da važi $A_{11} \in \mathbb{C}^{k\times k}$.
6: Izračunati vrednosti $R_1, \ldots, R_6$ i $X_{11}, X_{12}, X_{21}, X_{22}$ pomoću relacija (3.58) gde se inverzi računaju rekurzivno.
7: **return** $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$

3.5.2 Rekurzivna Cholesky faktorizacija

Za svaku Hermitsku, pozitivno definitnu matricu A postoji gornje trougaona matrica U takva da važi $A = U^*U$. Ovo je dobro poznata Cholesky faktorizacija matrice A. Ova faktorizacija se može generalisati i na singularne matrice. U radu [25] Courrieu je posmatrao ovu generalizaciju.

Teorema 3.5.3. [**25**] **(Courrieu 2002)** *Neka je A Hermitska (regularna ili singularna) pozitivno semi-definitna matrica reda $n \times n$. Tada postoji gornje trougaona matrica $U = [u_{ij}]$ takva da je $U^*U = A$ i $u_{ii} \geq 0$ za svako $i = 1, \ldots, n$. Ako za neki indeks i važi $u_{ii} = 0$, onda važi $u_{ij} = 0$ za svako $j = 1, \ldots, n$. Štaviše, matrica U je jedinstvena.*

Klasičan metod pivotiranja za računanje Cholesky faktorizacije može se veoma lako proširiti do metoda za računanje generalisane Cholesky faktorizacije. Takodje, i ova generalizacija je data u radu [24]. Algoritam 3.5.3 predstavlja tu generalizaciju.

Algoritam 3.5.3 Metod pivotiranja za računanje generalisane Cholesky faktorizacije [24]

Input: Pozitivno semi-definitna matrica $A \in \mathbb{C}^{n\times n}$.

1: **for** $i := 1$ to n **do**
2: $u_{ii} := \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ik}^2}$
3: **for** $j := i + 1$ to n **do**
4: $u_{ij} := \begin{cases} \left(a_{ij} - \sum_{k=0}^{i-1} u_{ik}u_{jk}\right)/u_{ii}, & u_{ii} \neq 0, \\ 0, & u_{ii} = 0 \end{cases}$
5: $u_{ji} := 0$
6: **end for**
7: **end for**
8: **return** $U = [u_{ij}]_{1\leq i,j\leq n}$

Pošto Algoritam 3.5.3 predstavlja generalizaciju poznatog metoda pivotiranja, njegova složenost je jednaka $\mathcal{O}(n^3)$.

U ovom odeljku razmotrićemo rekurzivni algoritam za računanje, najpre Cholesky, a zatim i generalisane Cholesky faktorizacije. Ovaj algoritam je u potpunosti blokovski i ima složenost

$\Theta(\text{mul}(n))$.

Pretpostavimo da je matrica A Hermitska i pozitivno definitna (a ujedno i regularna). Posmatrajmo ponovo blokovsku reprezentaciju (3.56) matrice A kao i odgovarajuću blokovsku reprezentaciju matrice U.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^* & A_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}, \quad U_{11}, A_{11} \in \mathbb{C}^{k\times k}. \tag{3.59}$$

Jednačina $A = U^*U$ je ekvivalentna sledećem sistemu matričnih jednačina

$$\begin{aligned} &1. \quad A_{11} = U_{11}^* U_{11} \\ &2. \quad A_{12} = U_{11}^* U_{12} \\ &3. \quad A_{22} = U_{12}^* U_{12} + U_{22}^* U_{22}. \end{aligned} \tag{3.60}$$

Regularni slučaj

Gustavson i Jonsson su u radu [50] prezentovali jedan metod za računanje Cholesky faktorizacije blok matrica. Drugi rezultati vezani za rekurzivne algoritme u linearnoj algebri mogu se pronaći na primer u radovima [49, 50, 143]. U rekurzivnom algoritmu iz rada [50], Cholesky faktorizacija pozitivno definitne Hermitske matrice izračunava se u tri koraka. Najpre se algoritam rekurzivno primeni na matricu A_{11} reda $n_1 = \lfloor n/2 \rfloor$. Nakon toga se matrica U_{12} izračunava rešavanjem $n_2 = \lceil n/2 \rceil$ trougaonih sistema jednačina (matrica svakog od sistema je dimenzija $n_1 = \lfloor n/2 \rfloor$). Na kraju se za matricu $\tilde{A}_{22} = A_{22} - U_{12}^* U_{12}$ rekurzivno izračuna generalisana Cholesky faktorizacija [50].

Složenost rešavanja $n \times n$ trougaonog sistema primenom metoda Gausove eliminacije je $\mathcal{O}(n^2)$. Prema tome, ukupna složenost rešavanja $n/2$ trougoaonih sistema dimenzija $n \times n$ je $\mathcal{O}(n^3)$.

Mi ćemo opisati alternativni metod za generisanje bloka U_{12}. Iz druge jednačine sistema (3.60) imamo da važi $U_{12} = (U_{11}^*)^{-1} A_{12}$. Činjenica da je matrica U_{11} (odnosno U_{11}^*) regularna sledi iz pozitivne definitnosti matrice A. Jasno je da je rešavanje $n_2 = \lceil n/2 \rceil$ trougaonih sistema linearnih jednačina ekvivalentno računanju izraza $(U_{11}^*)^{-1} A_{12}$. Zbog toga ćemo mi razviti metod za simultano rešavanje sistema (3.60) i računanje inverzne matrice $Y = U^{-1}$ koji je potpuno blokovski. Štaviše ovaj pristup ćemo primeniti i u slučaju generalisane Cholesky faktorizacije.

Posmatrajmo istu blok dekompoziciju matrice $Y = U^{-1}$ kao i za matricu U. Imamo da važi sledeća blokovska jednačina

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ 0 & Y_{22} \end{bmatrix} = \begin{bmatrix} U_{11}Y_{11} & U_{11}Y_{12} + U_{12}Y_{22} \\ 0 & U_{22}Y_{22} \end{bmatrix} = \begin{bmatrix} I_k & 0 \\ 0 & I_{n-k} \end{bmatrix}, \tag{3.61}$$

što je ekvivalentno sa

$$Y_{11} = U_{11}^{-1}, \quad Y_{22} = U_{22}^{-1}, \quad Y_{12} = -Y_{11} U_{12} Y_{22}. \tag{3.62}$$

Kombinovanjem jednačina (3.60) i (3.62), možemo rekurzivno računati matrice U i $Y = U^{-1}$. Na ovom principu je baziran Algoritam 3.5.4.

Algoritam 3.5.4 Potpuno rekurzivna Cholesky faktorizacija

Input: Regularna, Hermitska, pozitivno definitna matrica A formata $n \times n$.

if $n = 1$ **then**
  return $U := [\sqrt{a_{11}}]$, $Y := \left[\sqrt{a_{11}^{-1}}\right]$
end if
Izvršiti blok dekompoziciju $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ tako da važi $A_{11} \in \mathbb{C}^{k \times k}$, gde je $k = \lfloor \frac{n}{2} \rfloor$.
Izračunati rekurzivno Cholesky faktorizaciju matrice U_{11} i njenog inverza Y_{11} korišćenjem istog algoritma za ulaznu matricu A_{11}.
$U_{12} := Y_{11}^* A_{12}$
$T_1 := U_{12}^* U_{12}$
$T_2 := A_{22} - T_1$
Izračunati rekurzivno Cholesky faktorizaciju matrice U_{22} i njenog inverza Y_{22} korišćenjem istog algoritma za ulaznu matricu T_2.
$T_3 := -Y_{11} U_{12}$
$Y_{12} := T_3 Y_{22}$
return $U := \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$ i $Y := \begin{bmatrix} Y_{11} & Y_{12} \\ 0 & Y_{22} \end{bmatrix}$.

Propozicija 3.5.4. *Izlazne matrice U i Y iz Algoritma 3.5.4 zadovoljavaju $A = U^*U$ i $Y = U^{-1}$.*

Izračunajmo složenost Algoritma 3.5.4, koju ćemo označiti sa $\mathrm{Chol}(n)$. Sledeća teorema tvrdi da korišćenjem Strassenovog metoda za množenje matrica (ili bilo kog metoda koji ima složenost $\mathcal{O}(n^{2+\epsilon})$ gde je $0 < \epsilon < 1$), dobijamo da je složenost $\mathrm{Chol}(n)$ manja od složenosti metoda pivotiranja ($\mathcal{O}(n^3)$).

Teorema 3.5.5. *Pod pretpostavkom da je* $\mathrm{add}(n) = \mathcal{O}(n^2)$, $\mathrm{mul}(n) = \Theta(n^{2+\epsilon})$, *gde je* $0 < \epsilon < 1$, *složenost Algoritma 3.5.2 jednaka je*

$$\mathrm{Chol}(n) = \Theta(\mathrm{mul}(n)) = \Theta(n^{2+\epsilon}). \tag{3.63}$$

Dokaz. Ako odaberemo $k = \lfloor \frac{n}{2} \rfloor$, iz relacija (3.58) imamo da važi sledeći izraz za $\mathrm{Chol}(n)$, gde je $l = \lceil \frac{n}{2} \rceil$

$$\mathrm{Chol}(n) = \begin{cases} 1, & n = 1 \\ \mathrm{Chol}(k) + \mathrm{Chol}(l) + & \\ \mathrm{mul}(l,k,k) + \mathrm{mul}(k,k,l) + \mathrm{mul}(l,k,l) + \mathrm{mul}(k,l,l) & \\ \quad + \mathrm{mul}(l,l,k) + \mathrm{mul}(k,l,k) + \mathrm{add}(k) + \mathrm{add}(l), & n > 1. \end{cases}$$

Pretpostavimo da je n stepen dvojke. Tada, pošto je $k = l = n/2$ i $\mathrm{add}(n) = \mathcal{O}(n^2) < \mathrm{mul}(n)$, za $n > 1$ imamo da važi

$$\begin{aligned} \mathrm{Chol}(n) &= \mathrm{Chol}(n/2) + \mathrm{Chol}(n/2) + 6 \cdot \mathrm{mul}(n/2) \\ &= 2\mathrm{Chol}(n/2) + \Theta(\mathrm{mul}(n)). \end{aligned} \tag{3.64}$$

Pošto je $\text{mul}(n) = \Theta(n^{2+\epsilon})$, $0 < \epsilon < 1$, nije teško proveriti da važi $2 \cdot \text{mul}(n/2) < c \cdot \text{mul}(n)$, za neku konstantu $c < 1/2$. Prema tome, primenom slučaja 3 Master teoreme (videti na primer [23]) dobijamo rešenje rekurentne relacije (3.64) u obliku $\text{Chol}(n) = \Theta(\text{mul}(n))$ i dokazujemo (3.63).

Pretpostavimo sada da n nije stepen dvojke. Ako $U^*U = A$ važi, tada će za broj q takav da je $n + q$ najmanji stepen dvojke veći od n važiti

$$\begin{bmatrix} A & \mathbb{O} \\ \mathbb{O} & I_q \end{bmatrix} = \begin{bmatrix} U & \mathbb{O} \\ \mathbb{O} & I_q \end{bmatrix}^* \begin{bmatrix} U & \mathbb{O} \\ \mathbb{O} & I_q \end{bmatrix}.$$

Ovim mi proširujemo matricu do veličine koja je tačan stepen dvojke. Tada dobijamo da je tražena složenost jednaka $\text{Chol}(n) = \Theta(n^{2+\epsilon})$ uzimajući u obzir složenost $\Theta(\text{mul}(n+q))$ proširenog problema. Nije teško proveriti da važi $\text{mul}(n+q) = \mathcal{O}(\text{mul}(n))$ pošto je $\text{mul}(n) = \Theta(n^{2+\epsilon})$. Ova relacija garantuje da proširivanje neće promeniti složenost za više od konstantnog faktora. $\square$

Singularni slučaj

Uopštićemo Algoritam 3.5.4 na slučaj kada je ulazna Hermitska matrica A singularna ili pozitivno semi-definitna. Izlaz uopštenog algoritma biće generalisana Cholesky faktorizacija matrice A.

Primetimo da, ako je A singularna ili pozitivno semi-definitna matrica, kada se primeni Algoritam 3.5.4, tokom rada doći će do situacije da algoritam treba da izvrši korak 2 a da pritom važi $a_{11} = 0$. Nadalje, algoritam ne može više da nastavi sa radom.

Modifikovaćemo ovaj korak tako što ćemo u ovom slučaju vratiti na izlaz, takodje, nula matricu. Drugim rečima, ako je $n = 1$ i $A = [0]$ tada će algoritam vratiti $U = V = [0]$. Ova modifikacija je opisana u Algoritmu 3.5.5.

Algoritam 3.5.5 Potpuno rekurzivna generalisana Cholesky faktorizacija

Input: Hermitska, pozitivno semi-definitna matrica $A \in \mathbb{C}^{n\times n}$.

```
if n = 1 then
    if a_11 ≠ 0 then
        return U := [√a_11], Y := [√(a_11^{-1})]
    else
        return U := [0], Y := [0]
    end if
end if
Nastaviti sa koracima 4-12 Algoritma 3.5.4
```

Algoritam 3.5.5 vraća matricu U koja zadovoljava uslov $A = U^*U$, odnosno računa generalisanu Cholesky dekompoziciju. Takodje, izlazna matrica Y predstavlja $\{1, 2, 3\}$ inverz matrice U. Ovo je dokazano u sledećoj teoremi koja predstavlja glavni rezultat ovog poglavlja.

Teorema 3.5.6. *Posmatrajmo Hermitsku, pozitivnu semi-definitnu matricu $A \in R^{n\times n}$. Izlazna matrica U Algoritma 3.5.5 zadovoljava $A = U^*U$. Štaviše, izlazna matrica Y je $\{1,2,3\}$ inverz matrice U, matrica UY je dijagonalna pri čemu su elementi glavne dijagonale jednaki* 0 *ili* 1.

Dokaz.

Prvo primetimo da je svaki glavno-dijagonalni minor $A_{(S)}$, $S \subset \{1,\dots,n\}$ matrice A, takodje, pozitivno semi-definitan. Označimo sa $S^C = \{1,\dots,n\} \setminus S$ komplement skupa indeksa S. Neka je $x' \in \mathbb{C}^{|S|\times 1}$ proizvoljni vektor gde smo sa $|S|$ označili kardinalnost skupa S. Ako sada stavimo $x_{(S)} = x'$ i $x_{(S^C)} = 0$ dobijamo $0 \leq x^*Ax = x'^*A_{(S)}x'$. Ovim smo dokazali da je matrica $A_{(S)}$ pozitivno semi-definitna.

Dalje ćemo dokaz teoreme izvesti matematičkom indukcijom.

Za matrice tipa 1×1 teorema trivijalno važi na osnovu koraka 2 Algoritma 3.5.5. Pretpostavimo sada da tvrdjenje važi za sve matrice dimenzija manjih od n.

Da bi dokazali da važi $A = U^*U$ dovoljno je da dokažemo da su zadovoljene jednačine (3.60). Pošto je A pozitivno semi-definitna matrica, onda postoji matrica U' takva da je $U'^*U' = A$ (na osnovu Teoreme 3.5.3). Razložimo matricu U' na isti način kao matricu U u relaciji (3.59): $U' = \begin{bmatrix} U'_{11} & U'_{12} \\ 0 & U'_{22} \end{bmatrix}$. Tada matrica U' zadovoljava jednačine (3.60), tj. važi

$$\begin{aligned} &1. \quad A_{11} = U'^*_{11}U'_{11} \\ &2. \quad A_{12} = U'^*_{11}U'_{12} \\ &3. \quad A_{22} = U'^*_{12}U'_{12} + U'^*_{22}U'_{22}. \end{aligned} \tag{3.65}$$

U koraku 5 Algoritma 3.5.4, ovaj algoritam se rekurzivno primenjuje na matricu A_{11}. Pošto smo već dokazali da je A_{11} pozitivno semi-definitna matrica, iz induktivne hipoteze sledi da uslovi teoreme važe za matricu A_{11}, odnosno da je $U_{11}^*U_{11} = A_{11}$. Zbog jedinstvenosti matrice U'_{11} (ponovo na osnovu Teoreme 3.5.3) zaključujemo da važi $U'_{11} = U_{11}$. Ovim smo pokazali da važi prva jednačina u (3.60).

Iz druge jednačine u (3.65) imamo da je $U'^*_{11}U'_{12} = U^*_{11}U'_{12} = A_{12}$. Prema tome matrična jednačina $U^*_{11}X = A_{12}$ ima rešenja. Primenom Posledice 3.1.18 dobijamo da važi $U^*_{11}(U^{(1)}_{11})^*A_{12} = A_{12}$ za proizvoljni $\{1\}$ inverz $U^{(1)}_{11}$ matrice U_{11}. Iz induktivne hipoteze imamo da je $Y_{11} \in U_{11}\{1\}$, pa je onda

$$U^*_{11}Y^*_{11}A_{12} = A_{12}. \tag{3.66}$$

Na osnovu (3.66) i koraka 6 Algoritma 3.5.5, dobijamo da je

$$A_{12} = U^*_{11}Y^*_{11}A_{12} = U^*_{11}U_{12},$$

odnosno da važi druga jednačina u (3.60).

Da bi dokazali poslednju jednačinu u (3.60), potrebno je dokazati da je matrica $A_{22} - U^*_{12}U_{12}$ pozitivno semi-definitna. Neka je $y \in \mathbb{C}^{n-k\times 1}$ proizvoljan vektor. Stavimo $x_1 = -Y_{11}Y^*_{11}A_{12}y$,

$x_2 = y$ i $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Iz uslova pozitivne semi-definitnosti matrice A i $A_{11} = U_{11}^* U_{11}$ dobijamo

$$\begin{aligned} 0 \le x^* A x &= x_1^* A_{11} x_1 + 2 x_1^* A_{12} x_2 + x_2^* A_{22} x_2 \\ &= y^* A_{12}^* Y_{11} Y_{11}^* U_{11}^* U_{11} Y_{11} Y_{11}^* A_{12} y - 2 y^* A_{12}^* Y_{11} Y_{11}^* A_{12} y + y^* A_{22} y. \end{aligned}$$

Na osnovu induktivne hipoteze imamo da važi $Y_{11} \in A_{11}\{1,2,3\}$ i

$$U_{11} Y_{11} Y_{11}^* = (U_{11} Y_{11})^* Y_{11}^* = (Y_{11} U_{11} Y_{11})^* = Y_{11}^*. \tag{3.67}$$

Prema tome

$$\begin{aligned} 0 \le x^* A x &= y^* A_{22} y - y^* A_{12}^* Y_{11} Y_{11}^* A_{12} y \\ &= y^* (A_{22} - U_{12}^* U_{12}) y. \end{aligned} \tag{3.68}$$

Pošto $y^*(A_{22} - U_{12}^* U_{12}) y \ge 0$ važi za proizvoljni vektor $y \in R^{n-k \times n-k}$, matrica $A_{22} - U_{12}^* U_{12}$ je pozitivno semi-definitna. Iz induktivne hipoteze imamo da je $U_{22}^* U_{22} = A_{22} - U_{12}^* U_{12}$. Time smo dokazali da matrica U zadovoljava sistem (3.60), tj. da važi $U^* U = A$.

Preostaje nam da dokažemo da matrica Y pripada skupu $\{1,2,3\}$ inverza matrice U. Na osnovu Teoreme 1 rada [14] direktno dobijamo da matrica Y pripada skupu $\{2\}$ inverza matrice U. Možemo dokazati da je Y, takodje, i $\{1\}$ inverz matrice U direktnom proverom jednakosti $UYU = U$. Zaista, imamo da važi

$$UYU = \begin{bmatrix} U_{11} Y_{11} U_{11} & U_{11} Y_{11} U_{12} + U_{11} Y_{12} U_{22} + U_{12} Y_{22} U_{22} \\ \mathbb{O} & U_{22} Y_{22} U_{22} \end{bmatrix}. \tag{3.69}$$

Iz induktivne hipoteze dobijamo $U_{11} Y_{11} U_{11} = U_{11}$ i $U_{22} Y_{22} U_{22} = U_{22}$. Iz koraka 6 Algoritma 3.5.5 i (3.67) sledi

$$\begin{aligned} U_{11} Y_{11} U_{12} &= U_{11} Y_{11} Y_{11}^* A_{12} \\ &= Y_{11}^* A_{12} \\ &= U_{12}. \end{aligned} \tag{3.70}$$

Iz relacije (3.70) i definicije matrice Y_{12} u koraku 12 Algoritma 3.5.5 dobijamo

$$\begin{aligned} &U_{11} Y_{11} U_{12} + U_{11} Y_{12} U_{22} + U_{12} Y_{22} U_{22} \\ &\qquad = U_{12} - U_{11} Y_{11} U_{12} Y_{22} U_{22} + U_{12} Y_{22} U_{22} \\ &\qquad = U_{12} - U_{12} Y_{22} U_{22} + U_{12} Y_{22} U_{22} \\ &\qquad = U_{12}. \end{aligned} \tag{3.71}$$

Time smo dokazali da $Y \in U\{1\}$.

Da bi dokazali da je $Y \in U\{3\}$, potrebno je da dokažemo da je matrica UY Hermitska. Nije teško proveriti da važi

$$UY = \begin{bmatrix} U_{11} Y_{11} & U_{11} Y_{12} + U_{12} Y_{22} \\ \mathbb{O} & U_{22} Y_{22} \end{bmatrix}. \tag{3.72}$$

Ponovo na osnovu definicije matrica Y_{12}, U_{12} i Y_{22} u Algoritmu 3.5.5 dobijamo

$$\begin{aligned} U_{11} Y_{12} + U_{12} Y_{22} &= -U_{11} Y_{11} U_{12} Y_{22} + Y_{11}^* A_{12} Y_{22} \\ &= -U_{11} Y_{11} Y_{11}^* A_{12} Y_{22} + Y_{11}^* A_{12} Y_{22} = 0. \end{aligned} \tag{3.73}$$

Poslednja jednakost u (3.73), dokazuje se još jednom primenom svojstva $U_{11}Y_{11}Y_{11}^* = Y_{11}^*$. Prema tome dokazali smo da je $Y \in U\{1,2,3\}$. Iz relacija (3.72), (3.73) i induktivne hipoteze sada dobijamo da je matrica UY dijagonalna (samim tim i Hermitska) i da su joj elementi na glavnoj dijagonali jednaki 0 ili 1. $\square$

3.5.3 Algoritam za brzo računanje generalisanih inverza

U ovom odeljku pokazaćemo kako se rezultati dobijeni u predhodnim odeljcima mogu primeniti na računanje MP inverza kao i drugih klasa $\{i, j, \dots, k\}$ inverza. Glavni rezultat ovog odeljka su algoritmi za računanje MP i $\{i, j, \dots, k\}$ inverza čija je složenost jednaka $\Theta(\mathrm{mul}(n))$.

Koristićemo rezultate prikazane u radovima Courrieua [24] kao i Stanimirovića i Tasića [136]. P. Courrieu je u radu [24] iskoristio generalisanu Cholesky faktorizaciju matrice za računanje MP inverza.

Lema 3.5.7. [**24**] **(Courrieu 2005)** *Neka je A proizvoljna matrica formata $m \times n$ i neka je S^*S generalisana Cholesky faktorizacija matrice A^*A. Takodje, neka je matrica L takva da se L^* sastoji od vrsta matrice S koje su različite od nula vrste. Tada MP inverz matrice A zadovoljava sledeću relaciju*

$$A^\dagger = L(L^*L)^{-1}(L^*L)^{-1}L^*A^*. \tag{3.74}$$

Dokaz. Imamo da je matrica L potpunog ranga kolona. Stoga je matrica L^* potpunog ranga vrsta. Prema tome imamo da je $A^*A = LL^*$ faktorizacija potpunog ranga pa na osnovu Leme 3.1.12 i Teoreme 3.1.15 važi

$$\begin{aligned} A^\dagger &= (A^*A)^\dagger A^* = (LL^*)^\dagger A^* \\ &= L(L^*L)^{-1}(L^*L)^{-1}L^*A^*. \end{aligned}$$

Ovim je lema dokazana. $\square$

Kombinujući Algoritam 3.5.5 za računanje generalisane Cholesky faktorizacije, Algoritam 3.5.2 za rekurzivno računanje inverzne matrice i predhodnu lemu, dobijamo algoritam za računanje MP inverza u vremenu $\Theta(\mathrm{mul}(n))$ korišćenjem relacije (3.74).

Algoritam 3.5.6 Izračunavanje MP inverza u vremenu množenja matrica

Input: Matrica $A \in \mathbb{C}^{m \times n}$.

1: $A' := A^*A$
2: Naći generalisanu Cholesky faktorizaciju $A' = U^*U$ matrice A' primenom Algoritma 3.5.5.
3: Formirati matricu L^* izbacivanjem svih nula vrsta matrice U.
4: $T := L^*L$
5: Naći inverznu matricu $M := T^{-1}$ primenom Algoritma 3.5.2.
6: **return** $A^\dagger := LM^2L^*A^*$.

Da bi dokazali korektnost Algoritma 3.5.6 potrebno nam je još da dokažemo da su svi glavnodijagonalni minori matrice T, definisane u koraku 4 regularni.

Teorema 3.5.8. *Matrica $T = L^*L$ definisana u koraku 4 Algoritma 3.5.6 je Hermitska pozitivno definitna matrica i svi glavni dijagonalni minori ove matrice su regularni*

Dokaz. Neka je $T_{(S)}$ glavnodijagonalni minor matrice T definisan odgovarajućim skupom indeksa $S \subset \{1, \ldots, n\}$. Za proizvoljni nenula vektor $x \in \mathbb{C}^{|S|}$ važi

$$x^*T_{(S)}x = x^*L_{(S)}^*L_{(S)}x = (L_{(S)}x)^*L_{(S)}x.$$

Pošto je $L_{(S)}$ potpunog ranga kolona, imamo da je $L_{(S)}x \neq 0$, što dalje implicira $x'^*T_{(S)}x' > 0$. Prema tome matrica $T_{(S)}$ je pozitivno definitna, a samim tim i regularna. $\square$

Svi koraci Algoritma 3.5.6 rade u vremenu $\Theta(\text{mul}(n))$, pa je, prema tome, to ujedno i složenost Algoritma 3.5.6. Ukoliko se za množenje matrica koristi neki od algoritama koji radi u vremenu $\mathcal{O}(n^{2+\epsilon})$ gde je $0 < \epsilon < 1$, tada je to ujedno i složenost Algoritma 3.5.6.

U radu [136] (Teorema 2.1), Stanimirović i Tasić su su dali uopštenje Courrieuovog metoda za računanje raznih klasa $\{i, j, \ldots, k\}$ inverza, uključujući $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{2, 3\}_s$ i $\{2, 4\}_s$ inverze.

Teorema 3.5.9. **[136] (Stanimirović, Tasić 2007)** *Neka je $A \in \mathbb{C}_r^{m\times n}$ proizvoljna matrica. Neka je $0 < s \leq r$ proizvoljan prirodni broj i neka su m_1, n_1 prirodni brojevi takvi da je $m_1, n_1 \geq s$. Tada važi:*

(a) $A\{2,4\}_s = \{L(L^*L)^{-2}L^*(R^*A)^*R^* |\ R \in \mathbb{C}_s^{m\times n_1}, R^*A \in \mathbb{C}_s^{n_1\times n}\}$, *gde je* $(R^*A)^*(R^*A) = LL^*$ *generalisana Cholesky faktorizacija pri čemu su iz matrice L izbačene nula kolone.*

(b) $A\{2,3\}_s = \{T^*(AT^*)^*L(L^*L)^{-2}L^* |\ T \in \mathbb{C}_s^{m_1\times n}, AT^* \in \mathbb{C}_s^{m\times m_1}\}$, *gde je* $(AT^*)(AT^*)^* = L^*L$ *generalisana Cholesky faktorizacija pri čemu su iz matrice L izbačene nula kolone.*

(c) $A\{1,2,4\} = \{L(L^*L)^{-2}L^*(R^*A)^*R^* |\ R \in \mathbb{C}_r^{m\times n_1}, R^*A \in \mathbb{C}_r^{n_1\times n}\}$, *gde je* $(R^*A)^*(R^*A) = LL^*$ *generalisana Cholesky faktorizacija pri čemu su iz matrice L izbačene nula kolone.*

(d) $A\{1,2,3\} = \{T^*(AT^*)^*L(L^*L)^{-2}L^* |\ T \in \mathbb{C}_r^{m_1\times n}, AT^* \in \mathbb{C}_r^{m\times m_1}\}$, *gde je* $(AT^*)(AT^*)^* = LL^*$ *generalisana Cholesky faktorizacija pri čemu su iz matrice L izbačene nula kolone.*

Kombinovanjem rezultata iz predhodnih odeljaka zajedno sa teoremom 3.5.9, dobijamo metod za računanje svih pomenutih klasa $\{i, j, \ldots, k\}$ inverza u vremenu $\Theta(\text{mul}(n))$. Ovaj metod je realizovan Algoritmom 3.5.7. Primetimo da Algoritam 3.5.7 opisuje dva analogna metoda (drugi se dobija kada se odgovarajući izrazi zamene izrazima u zagradama).

Sledeća teorema dokazuje korektnost Algoritma 3.5.7.

Teorema 3.5.10. *Posmatrajmo proizvoljnu matricu $A \in \mathbb{C}_r^{m\times n}$. Neka je $0 < s \leq r$ proizvoljan prirodni broj i neka su m_1 i n_1 prirodni brojevi takvi da je $m_1, n_1 \geq s$. Tada za izlaznu matricu Algoritma 3.5.7 važi:*

Algoritam 3.5.7 Izračunavanje $\{i, j, \ldots, k\}$ inverza u vremenu množenja matrica

Input: Matrica $A \in \mathbb{C}_r^{m\times n}$ kao i matrica $R \in \mathbb{C}_s^{m\times n_1}$ ($T \in \mathbb{C}_s^{m_1\times n}$), gde je $0 < s \leq r$ i $m_1 \geq s$ ($n_1 \geq s$).

1: $P := (AT^*)(AT^*)^*$ ($Q := (R^*A)^*(R^*A)$)
2: Naći generalisanu Cholesky faktorizaciju $P := U^*U$ ($Q := U^*U$), primenom Algoritma 3.5.5.
3: Formirati matricu L^* izbacivanjem nula vrsta iz matrice U.
4: $T := L^*L$
5: Naći inverznu matricu $M := T^{-1}$ primenom Algoritma 3.5.2
6: **return** $X_M := LM^2L^*A^*RR^*$ ($X_N := T^*TA^*LM^2L^*$).

(a) *Ako je* $s < r$, $X_M \in A\{2,4\}_s$.

(b) *Ako je* $s < r$, $X_N \in A\{2,3\}_s$.

(c) *Ako je* $s = r$, $X_M \in A\{1,2,4\}$.

(d) *Ako je* $s = r$, $X_N \in A\{1,2,3\}$.

(e) *U slučaju* $R = A$ ($T = A$) *inverz* X_M *dobijen u delu* **(c)** *(inverz* X_N *dobijen u delu* **(d)***) jednak je* $A^\dagger$.

Štaviše, Algoritam 3.5.7 radi u vremenu $\Theta(\mathrm{mul}(n))$, *ako je* $\mathrm{mul}(n) = \mathcal{O}(n^{2+\epsilon})$ *gde je* $0 < \epsilon < 1$.

Dokaz. Matrice P i Q su Hermitske pozitivno definitne matrice. Prema tome, ulazna matrica za Algoritam 3.5.5 u koraku 2 zadovoljava potrebne uslove. Primenjujući isti postupak kao u dokazu Teoreme 3.5.8 i ovde možemo dokazati da je matrica $T = L^*L$ regularna i da isto važi za sve njene glavnodijagonalne minore. Ovo znači da je ulazna matrica u Algoritam 3.5.2 u koraku 5 korektna.

Uslovi **(a)**–**(e)** slede iz Teoreme 3.5.9.

Koristeći činjenicu da se inverzna matrica računa u vremenu $\Theta(\mathrm{mul}(n))$ kao i Teoremu 3.5.5, dobijamo da je složenost svih koraka Algoritma 3.5.7 jednaka $\Theta(\mathrm{mul}(n))$. □

Na osnovu svih do sada izloženih rezultata u ovom poglavlju možemo da konstatujemo da je moguće izračunati kako MP inverz tako i sve $\{1,2,3\}$, $\{1,2,4\}$, $\{2,3\}$ i $\{2,4\}$ inverze u vremenu množenja matrica, tj. u vremenu ispod $\mathcal{O}(n^3)$.

3.5.4 Rezultati testiranja i primeri

Svi algoritmi su implementirani u simboličkom programskom paketu MATHEMATICA. Kodovi najvažnijih funkcija dati su u sledećem odeljku. Napomenimo da se množenje matrica u MATHEMATICA-i izvršava za vreme $\mathcal{O}(n^3)$, pa je ovo ujedno i složenost odgovarajućih implementacija algoritama.

Primer 3.5.1. Pokazaćemo kako Algoritam 3.5.6 i Algoritam 3.5.7 rade na sledećem primeru

$$A = \begin{bmatrix} 8 & 8 & 1 & 1 \\ 10 & 10 & 2 & 2 \\ 11 & 11 & 3 & 3 \\ 12 & 12 & 4 & 4 \end{bmatrix}.$$

Primenom Algoritma 3.5.5 na matricu A^*A dobijamo generalisanu Cholesky faktorizaciju matrice A^*A

$$A^*A = U^*U \text{ where } U = \begin{bmatrix} \sqrt{429} & \sqrt{429} & \frac{109}{\sqrt{429}} & \frac{109}{\sqrt{429}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{\frac{989}{429}} & \sqrt{\frac{989}{429}} \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Generalisani inverz Y matrice U koji vraća Algoritam 3.5.5 jednak je

$$Y = \begin{bmatrix} \frac{1}{\sqrt{429}} & 0 & -\frac{109}{\sqrt{424281}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{\frac{429}{989}} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Proverom možemo utvrditi da je to zaista $\{1, 2, 3\}$ inverz matrice U. Ovo se slaže sa $UY = \mathrm{diag}(1, 0, 1, 0)$. Dobijeni rezultati su u skladu sa Teoremom 3.5.6. Iako matrica Y ne zadovoljava jednačinu (4) sistema (3.2), imamo da je matrica YU gornje trougaona sa samo nekoliko nenula vrednosti iznad glavne dijagonale

$$YU = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Sledeći korak je formiranje matrice L^* izbacivanjem nula vrsta iz matrice U i invertovanje matrice L^*L korišćenjem Algoritma 3.5.2. Imamo da važi

$$L = \begin{bmatrix} \sqrt{429} & 0 \\ \sqrt{429} & 0 \\ \frac{109}{\sqrt{429}} & \sqrt{\frac{989}{429}} \\ \frac{109}{\sqrt{429}} & \sqrt{\frac{989}{429}} \end{bmatrix}.$$

Dalje, računamo matricu M na sledeći način

$$L^*L = \begin{bmatrix} \frac{391844}{429} & \frac{218\sqrt{989}}{429} \\ \frac{218\sqrt{989}}{429} & \frac{1978}{429} \end{bmatrix} \Longrightarrow M = (L^*L)^{-1} = \begin{bmatrix} \frac{1}{858} & -\frac{109}{858\sqrt{989}} \\ -\frac{109}{858\sqrt{989}} & \frac{97961}{424281} \end{bmatrix}.$$

Konačno dobijamo traženi MP inverz primenjujući poslednji korak Algoritma 3.5.6

$$A^\dagger = \begin{bmatrix} \frac{131}{1978} & \frac{41}{989} & \frac{3}{1978} & -\frac{38}{989} \\ \frac{131}{1978} & \frac{41}{989} & \frac{3}{1978} & -\frac{38}{989} \\ -\frac{443}{1978} & -\frac{116}{989} & \frac{44}{989} & \frac{204}{989} \\ -\frac{443}{1978} & -\frac{116}{989} & \frac{44}{989} & \frac{204}{989} \end{bmatrix}.$$

Primer 3.5.2. Za matrice

$$A = \begin{bmatrix} 30 & 78 & 54 & 66 & 66 & 42 & 60 \\ 42 & 89 & 55 & 70 & 82 & 51 & 74 \\ 78 & 113 & 34 & 55 & 127 & 66 & 98 \\ 96 & 115 & 80 & 113 & 137 & 108 & 166 \end{bmatrix}, \mathrm{rang}(A) = 3$$

$$R = \begin{bmatrix} 52 & 9 & 23 & 40 & 35 & 5 & 37 \\ 92 & 54 & 72 & 64 & 56 & 30 & 68 \\ 4 & 18 & 16 & 0 & 0 & 10 & 4 \\ 22 & 54 & 51 & 8 & 7 & 30 & 19 \end{bmatrix}, \mathrm{rang}(R) = 2$$

primenom Algoritma 3.5.7 dobijamo sledeći $\{2, 4\}$ inverz matrice A ranga 2

$$A_2^{(2,4)} = \begin{bmatrix} -\frac{2064786876}{231354215041} & -\frac{4755131732}{694062645123} & \frac{1424383465}{694062645123} & \frac{5928345641}{694062645123} \\ \frac{4016182158}{231354215041} & \frac{3180731937}{231354215041} & -\frac{1034331045}{462708430082} & -\frac{6217922757}{462708430082} \\ \frac{2682758156}{231354215041} & \frac{6419265925}{694062645123} & -\frac{848359916}{694062645123} & -\frac{5890267708}{694062645123} \\ -\frac{2365817440}{231354215041} & -\frac{1833498584}{231354215041} & \frac{471776128}{231354215041} & \frac{2133596416}{231354215041} \\ -\frac{2070090260}{231354215041} & -\frac{1604311261}{231354215041} & \frac{412804112}{231354215041} & \frac{1866896864}{231354215041} \\ \frac{2231212310}{231354215041} & \frac{5301219895}{694062645123} & -\frac{1723885075}{1388125290246} & -\frac{10363204595}{1388125290246} \\ -\frac{1177605336}{231354215041} & -\frac{2692445825}{694062645123} & \frac{893635321}{694062645123} & \frac{3528049673}{694062645123} \end{bmatrix}.$$

Implementacije smo testirali na nekoliko slučajno generisanih test primera.

U prvim trima tabelama uporedjivali smo performanse implementacija Algoritma 3.5.5 (funkcija `Ch`) kao i Algoritma 3.5.3 (metod pivotiranja, funkcija `Cholesky`). Sva data vremena su u sekundama i dobijena su usrednjavanjem vremena izvršenja na ukupno 20 različitih slučajno generisanih test matrica istog ranga i dimenzija. Dimenzije matrica su 2^k za $k = 4, 5, 6, 7$ kao i vrednosti $\lfloor 2^k \sqrt{2} \rfloor$, takodje, za $k = 4, 5, 6, 7$.

n	`Ch`	`Cholesky`[136]
16	0.010	0.
23	0.011	0.005
32	0.018	0.016
45	0.021	0.047
64	0.052	0.120
90	0.078	0.328
128	0.151	0.905
180	1.295	3.562

rank$A = n$

n	`Ch`	`Cholesky`[136]
16	0.0047	0.0031
23	0.016	0.023
32	0.0256	0.0171
45	0.031	0.047
64	0.042	0.1232
90	0.078	0.328
128	0.1498	0.9207
180	0.265	2.481
256	3.24	10.5428
362	8.83	348.6
512	25.9895	1959.15

$\text{rank}A = n/2$

n	`Ch`	`Cholesky`[136]
16	0.0	0.0
23	0.003	0.0062
32	0.0094	0.0188
45	0.0094	0.0436
64	0.0378	0.1308
90	0.0346	0.3306
128	0.078	0.9422
180	0.195	2.5895
256	0.5616	7.9872
362	1.17	197.622
512	12.32	983.4

$\text{rank}A = n/10$

U naredne tri tabele poredili smo implementacije Algoritma 3.5.6 sa odgovarajućom implementacijom Algoritma 2.1 iz rada [136]. Ovi algoritmi izračunavaju MP inverz.

n	Alg. 3.5.6	Alg. 2.1 [136]
16	0.00775	0.
23	0.016	0.0155
32	0.031	0.01575
45	0.039	0.0505
64	0.05025	0.133
90	0.09725	0.3315
128	0.17125	0.94
180	0.31175	2.62075

$\text{rank}A = n$

n	Alg. 3.5.6	Alg. 2.1 [136]
16	0.	0.00375
23	0.01575	0.
32	0.01575	0.0115
45	0.02325	0.03525
64	0.043	0.08175
90	0.08975	0.2225
128	0.15225	0.7645
180	0.2925	2.57

$\text{rank}A = n/2$

n	Alg. 3.5.6	Alg. 2.1 [136]
16	0.00775	0.
23	0.	0.004
32	0.	0.0155
45	0.00775	0.0115
64	0.0235	0.01575
90	0.043	0.03125
128	0.1015	0.06625
180	0.17175	0.164

$\text{rank}A = n/10$

Konačno u poslednjim trima tabelama poredili smo implementacije Algoritma 3.5.7 i Algoritma 2.1 iz [136] kada oni računaju $\{2,3\}$ i $\{2,4\}$ inverze.

n	Alg. 3.5.7	Alg. 3.5.5
16	0.0115	0.008
23	0.0195	0.00775
32	0.02375	0.02325
45	0.03125	0.04675
64	0.05075	0.1365
90	0.09375	0.35125
128	0.17175	0.98675
180	0.328	2.699

$\text{rank}A = \text{rank}R = n$

n	Alg. 3.5.7	Alg. 3.5.5
16	0.0115	0.004
23	0.0195	0.008
32	0.02725	0.01575
45	0.03125	0.05475
64	0.0545	0.1405
90	0.0935	0.35475
128	0.16425	0.97875
180	0.30025	2.777

$\text{rank}A = \text{rank}R = n/2$

n	Alg. 3.5.7	Alg. 3.5.5
16	0.	0.
23	0.02725	0.01175
32	0.0195	0.02725
45	0.0315	0.0505
64	0.05475	0.14025
90	0.09775	0.38575
128	0.152	1.09975
180	0.28075	2.96

$\text{rank}A = \text{rank}R = n/10$

Rezultati nesumnjivo potvrdjuju da su naše implementacije bolje u skoro svim primerima. Performanse funkcije `Ch` kao i funkcija koje implementiraju Algoritam 3.5.6 i Algoritam 3.5.7 su dodatno degradirane zbog sporih rekurzivnih poziva u programskom paketu MATHEMATICA. Prema tome, možemo da zaključimo da su algoritmi koje smo uveli u ovom poglavlju bolji od odgovarajućih klasičnih algoritama čak i u slučaju kada je složenost i jednih i drugih $\mathcal{O}(n^3)$.

3.5.5 Implementacioni detalji i kodovi

Algoritam 3.5.5 za računanje generalisane Cholesky faktorizacije implementiran u sledećoj funkciji napisanoj u programskom paketu MATHEMATICA.

```
Ch[AA_] :=
    Module[{A, U11, U11inv, U22inv, U12, U22, A11, A12, A21, A22,
            Uinv, U, m, n, m1p, n1p, n1},
      A = AA;
      {m, n} = Dimensions[A];
      A = Chop[A, 10^(-6)];
      If [Simplify[A] == 0*A, Return[{0*A, 0*A}]];
      If [n == 1,
        If[A[[1, 1]] < 0, Return[{0*A, 0*A}]];
        Return[{{{Sqrt[A[[1, 1]]]}}, {{1/Sqrt[A[[1, 1]]]}}}]
        ];
      n1 = n/2 // Floor;
      m1p = n1p = n1;

      A11 = A // Take[#, m1p] & // Transpose[Take[Transpose[#], n1p]] &;
      A12 = A // Take[#, m1p] & // Transpose[Drop[Transpose[#], n1p]] &;
```

```
      A21 = A // Drop[#, m1p] & // Transpose[Take[Transpose[#], n1p]] &;
      A22 = A // Drop[#, m1p] & // Transpose[Drop[Transpose[#], n1p]] &;

      {U11, U11inv} = Ch[A11];
      If [Simplify[U11] == 0*U11,
        U12 = 0*A12;,
        U12 = Conjugate[Transpose[U11inv]].A12;
      ];
      {U22, U22inv} = Ch[A22 - Conjugate[Transpose[U12]].U12];
      U =
        Join[Transpose[Join[Transpose[U11], Transpose[U12]]],
          Transpose[Join[Transpose[0*A21], Transpose[U22]]]];
      Uinv =
        Join[Transpose[
            Join[Transpose[U11inv], Transpose[-U11inv.U12.U22inv]]],
          Transpose[Join[Transpose[0*A21], Transpose[U22inv]]]];
      Return[{U, Uinv} // Simplify]
    ];
```

Funkcija `Adop` je pomoćna funkcija i implementira korak 3 Algoritma 3.5.6 i Algoritma 3.5.7.

```
Adop[a_List] := Module[{m, n, a1, i},
    {m, n} = Dimensions[a];
    a1 = Transpose[a];
    Do[
      If [Chop[Norm[Abs[a1[[i]]]], 10^(-6)] == 0,
        a1 = Drop[a1, {i}];
        ];
      , {i, Length[a1], 1, -1}
      ];
    Return[Transpose[a1]];
  ];
```

Sledeće funkcije implementiraju redom Algoritam 3.5.6 (funkcija `MoorePenroseCh`) i Algoritam 3.5.7 (funkcije `Inverse1Ch` i `Inverse2Ch`).

```
MoorePenroseCh[A_] := Module[{U, Uinv, M},
    {U, Uinv} = Ch[Transpose[A].A];
    U = Adop[Conjugate[Transpose[U]]];
    M = Inverse[Conjugate[Transpose[U]].U];
    Return[U.M.M.Conjugate[Transpose[U]].Conjugate[Transpose[A]]];
  ];

Inverse1Ch[A_, R_] := Module[{A1, U, Uinv, M},
    A1 = Transpose[R].A;
    {U, Uinv} = Ch[Transpose[A1].A1];
    U = Adop[Conjugate[Transpose[U]]];
    M = Inverse[Conjugate[Transpose[U]].U];
    Return[U.M.M.Conjugate[Transpose[U]].Conjugate[Transpose[A1]]
    .Conjugate[Transpose[R]] // Simplify];
  ];

Inverse2Ch[A_, T_] := Module[{A1, U, Uinv, M},
    A1 = A.Transpose[T];
```

```
  {U, Uinv} = Ch[A1.Transpose[A1]];
  U = Adop[Conjugate[Transpose[U]]];
  M = Inverse[Conjugate[Transpose[U]].U];
  Return[Conjugate[Transpose[T]].Conjugate[Transpose[A1]].U.M.M
  .Conjugate[Transpose[U]] // Simplify];
];
```

Glava 4

Generalisani inverzi racionalnih i polinomijalnih matrica i primene

U ovoj glavi proučićemo algoritme za računanje generalisanih inverza polinomijalnih i racionalnih matrica. Na kraju glave ukazaćemo na neke primene teorije generalisanih inverza kako konstantnih tako i polinomijalnih i racionalnih matrica.

Ova glava sadrži naše originalne rezultate od kojih je velika većina publikovana u našim radovima [100, 101, 102, 103, 104, 130, 141].

4.1 Racionalne i polinomijalne matrice

U ovom poglavlju ćemo se podsetiti nekih svojstava racionalnih i polinomijalnih matrica koja ćemo u nastavku koristiti. Neka je $\mathbb{F}$ proizvoljno polje. U ovoj glavi ćemo se isljučivo baviti poljima realnih i kompleksnih brojeva, $\mathbb{R}$ i $\mathbb{C}$. Nezavisno od toga, sve definicije i svojstva koja ćemo u ovom poglavlju formulisati važe (odnosno imaju smisla) i za proizvoljno polje $\mathbb{F}$.

Matrica $A = [a_{ij}]$ formata $m \times n$ je *polinomijalna* (*racionalna*) ako su svi njeni elementi a_{ij} polinomi (racionalne funkcije) promenljive s. Racionalne, odnosno polinomijalne matrice označavaćemo sa $A(s)$. Na sličan način možemo definisati racionalne i polinomijalne matrice više promenljivih, koje označavamo sa $A(s_1, \ldots, s_p)$ (ako su promenljive označene sa $s_1, \ldots, s_p$). U ovom slučaju, koristimo kraću notaciju $A(S)$ gde je $S = (s_1, \ldots, s_p)$.

Skup svih polinomijalnih (racionalnih) matrica formata $m \times n$ označićemo sa $\mathbb{F}^{m\times n}[s]$ ($\mathbb{F}^{m\times n}(s)$). Sličnu notaciju koristimo i za racionalne ($\mathbb{F}^{m\times n}(s_1, \ldots, s_p) = \mathbb{F}^{m\times n}(S)$) odnosno polinomijalne ($\mathbb{F}^{m\times n}[s_1, \ldots, s_p] = \mathbb{F}^{m\times n}[S]$) matrice više promenljivih.

Sledeće definicije uvode pojam stepena polinomijalne matrice $A(s)$.

Definicija 4.1.1. *Za datu polinomijalnu matricu $A(s) \in \mathbb{F}[s]^{n\times m}$ definišemo njen maksimalni stepen (stepen) kao maksimalni stepen svih njenih elemenata*

$$\deg A(s) = \max\{\mathrm{dg}(A(s))_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}. \tag{4.1}$$

Na sličan način definišemo maksimalni stepen po promenljivoj k, $\deg_k A(S)$ polinomijalne matrice $A(s_1, \ldots, s_p)$ više promenljivih

$$\deg_k A(S) = \max\{\mathrm{dg}_k(A(S))_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}, \tag{4.2}$$

gde je $\mathrm{dg}_k P(S)$ stepen promenljive s_k u polinomu $P(S)$.

Definicija 4.1.2. *Matrica stepena polinomijalne matrice $A(s) \in \mathbb{F}[s]^{n\times m}$ je matrica definisana sa $\mathrm{dg}A(s) = [\mathrm{dg}A(s)_{ij}]_{m\times n}$.*

Sledeća lema daje neka osnovna svojstva matrica stepena.

Lema 4.1.1. *Neka su $A(s), B(s) \in \mathbb{F}^{n\times n}[s]$ i neka je $a(s) \in \mathbb{F}[s]$. Važe sledeće činjenice*

(a) $\mathrm{dg}(A(s)B(s))_{ij} = \max\{\mathrm{dg}A(s)_{ik} + \mathrm{dg}B(s)_{kj} \mid 1 \leq k \leq n\}$.

(b) $\mathrm{dg}(A(s) + B(s))_{ij} \leq \max\{\mathrm{dg}A(s)_{ij}, \mathrm{dg}B(s)_{ij}\}$.

(c) $\mathrm{dg}(a(s)A(s))_{ij} = \mathrm{dg}A(s)_{ij} + \mathrm{dg}(a(s))$.

Dokaz.
(a) Iz definicije proizvoda dve matrice i koristeći jednostavne formule

$$\mathrm{dg}(p(s) + q(s)) \leq \max\{\mathrm{dg}(p(s)), \mathrm{dg}(q(s))\}, \quad \mathrm{dg}(p(s)q(s)) = \mathrm{dg}(p(s)) + \mathrm{dg}(q(s))$$

koje važe za svako $p(s), q(s) \in \mathbb{F}(s)$ možemo da zaključimo da je

$$\mathrm{dg}(A(s)B(s))_{ij} = \mathrm{dg}((A(s)B(s))_{ij}) \leq \max\{\mathrm{dg}A(s)_{ik} + \mathrm{dg}B(s)_{kj} \mid k = 1, \ldots, n\}.$$

Ovim smo dokazali deo **(a)**. Naredna dva dela se dokazuju analogno. □

Svaka polinomijalna matrica $A(s) \in \mathbb{F}^{m\times n}[s]$ može da se napiše u obliku

$$A(s) = A_0 + A_1 s + \ldots + A_d s^d,$$

gde je $d = \deg A(s)$. Matrice $A_i \in \mathbb{F}^{m\times n}$ zvaćemo *koeficijent matrice.* Na sličan način možemo predstaviti polinomijalnu matricu više promenljivih $A(s_1, \ldots, s_p) \in \mathbb{F}^{m\times n}[s_1, \ldots, s_p]$ u obliku

$$A(S) = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \cdots \sum_{i_p=0}^{d_p} A_{i_1 i_2 \cdots i_p} s_1^{i_1} s_2^{i_2} \cdots s_p^{i_p} = \sum_{I=0}^{D} A_I S^I.$$

Ovde smo označili $D = (d_1, \ldots, d_p)$, $I = (i_1, \ldots, i_p)$, $S^I = s_1^{i_1} s_2^{i_2} \cdots s_p^{i_p}$, $A_I = A_{i_1 i_2 \cdots i_p} \in \mathbb{F}^{m\times n}$ kao i $\sum_{I=0}^{D} = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \cdots \sum_{i_p=0}^{d_p}$.

Neformalno, rećićemo da je matrica retka ukoliko ima malo elemenata koji su različiti od nule. Na sličan način, kažemo da je polinom redak ukoliko ima malo koeficijenata različitih od nule. Sledećim dvema definicijama formalno uvodimo veličine kojima određujemo retkost (sparsity) matrice.

Definicija 4.1.3. *Za datu polinomijalnu matricu $A(s) \in \mathbb{R}^{m\times n}[s]$,* **prvi spars broj** $sp_1(A)$ *matrice A je odnos ukupnog broja nenula elemenata u $A(s)$ i ukupnog broja elemenata, tj.*

$$sp_1(A(s)) = \frac{\#\,\{(i,j)\,|\,a_{ij}(s) \neq 0\}}{m\cdot n}.$$

Na isti način se prvi spars broj uvodi i za konstantne matrice

Prvi spars broj predstavlja gustinu nenula elemenata i ima vrednosti između 0 i 1

Definicija 4.1.4. *Za datu nekonstantnu polinomijalnu matricu $A(s) \in \mathbb{F}[s]^{m\times n}$,* **drugi spars broj** $sp_2(A(s))$ *je odnos*

$$sp_2(A(s)) = \frac{\#\left\{(i,j,k)\,|\,[s^k]a_{ij}(s) \neq 0\right\}}{(1+\deg A(s))\cdot m\cdot n}.$$

Drugi spars broj predstavlja gustinu nenula koeficijenata sadržanih u elementima $a_{ij}(s)$ matrice $A(s)$, i takođe ima vrednosti između 0 i 1.

Ilustrovaćemo računanje spars brojeva $sp_1(A(s))$ i $sp_2(A(s))$ na jednom primeru.

Primer 4.1.1. Razmotrimo sledeću polinomijalnu matricu

$$A(s) = \begin{bmatrix} s & s^2 & 1+s+s^2 \\ 0 & 0 & 1 \\ 1 & s^2 & 0 \end{bmatrix}.$$

Imamo da važi

$$\begin{aligned} \#\,\{(i,j)\mid a_{ij}(s)\neq 0\} =& \#\{(1,1),(1,2),(1,3),(2,3),(3,1),(3,2)\} = 6, \\ \#\left\{(i,j,k)\mid [S^k]a_{ij}(s)\neq 0\right\} =& \#\{(1,1,1),(1,2,2),(1,3,0),(1,3,1), \\ & (1,3,2),(2,3,0),(3,1,0),(3,2,2)\} = 8. \end{aligned}$$

Prema tome, spars brojevi $sp_1(A(s))$ i $sp_2(A(s))$ jednaki su

$$sp_1(A(s)) = \frac{6}{3\cdot 3} = \frac{2}{3}, \qquad sp_2(A(s)) = \frac{8}{3\cdot 3\cdot 2} = \frac{4}{9}.$$

Spars brojevi mogu se analogno definisati i za polinomijalne matrice više promenljivih.

Definicija 4.1.5. *Za datu polinomijalnu matricu više promenljivih $A(S) = [a_{ij}(S)] \in \mathbb{C}[S]^{m\times n}$ i $S = (s_1,\ldots,s_p)$,* **prvi spars broj** $sp_1(A)$ *je odnos ukupnog broja nenula elemenata i ukupnog broja elemenata u $A(S)$, tj.*

$$sp_1(A(S)) = \frac{\#\,\{(i,j)\mid a_{ij}(S)\neq 0\}}{m\cdot n}.$$

Definicija 4.1.6. *Za datu nekonstantnu polinomijalnu matricu više promenljivih $A(S) \in \mathbb{C}[S]^{m\times n}$ i $S = (s_1,\ldots,s_p)$,* **drugi spars broj** $sp_2(A(S))$ *jednak je sledećem odnosu*

$$sp_2(A(S)) = \frac{\#\{(i,j,k_1,\ldots,k_p)\mid 0\leq k_j\leq \deg_{s_j} A(S), [s_1^{k_1}\cdots s_p^{k_p}]a_{ij}(S)\neq 0\}}{(\deg_{s_1} A(S)+1)\cdots(\deg_{s_p} A(S)+1)\cdot m\cdot n}.$$

Primer 4.1.2. Posmatrajmo sledeću polinomijalnu matricu od dve promenljive $A(s_1, s_2)$

$$A(s_1, s_2) = \begin{bmatrix} s_1 + s_2 & s_1^2 & s_2^2 \\ s_1 s_2 & 2s_1 & 1 + s_1^2 \\ 0 & 0 & 1 \end{bmatrix}.$$

Imamo da je $\deg_1 A(s_1, s_2) = \deg_2 A(s_1, s_2) = 2$, a takođe važi

$$\begin{aligned} \#\{(i,j) \mid a_{ij}(s_1, s_2) \neq 0\} =& \#\{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,3)\} = 7, \\ \#\left\{(i,j,k) \mid [s^k]a_{ij}(s) \neq 0\right\} =& \#\{(1,1,1,0),(1,1,0,1),(1,2,2,0),(1,3,0,2),(2,1,1,1), \\ & (2,2,1,0),(2,3,0,0),(2,3,1,0),(2,3,2,0),(3,3,0,0)\} = 10. \end{aligned}$$

Iz predhodne relacije sada lako računamo spars brojeve

$$sp_1(A(s)) = \frac{7}{3 \cdot 3} = \frac{7}{9}, \qquad sp_2(A(s)) = \frac{10}{3 \cdot 3 \cdot 2 \cdot 2} = \frac{5}{18}.$$

4.2 Izračunavanje Moore-Penroseovog inverza polinomijalnih matrica interpolacijom

U ovom poglavlju konstruisaćemo interpolacioni metod za izračunavanje MP inverza date polinomijalne matrice, zasnovanog na Leverrier-Faddeevom metodu. Metod za procenu matrica stepena, takodje nastao iz Leverrier-Faddeevog metoda je dat kao poboljšanje interpolacionog metoda. Metodi su formulisani u obliku algoritma, implementirani u simboličkom programskom jeziku MATHEMATICA i testirani na nekoliko različitih klasa test primera.

Ovo poglavlje predstavlja naše originalne rezultate i zasnovano je na našem radu [130].

Dodajmo da u ovom i u sledeća dva poglavlja pretpostavljamo da su elementi polazne matrice A polinomi sa **realnim** koeficijentima.

4.2.1 Leverrier-Faddeev metod za izračunavanje Moore-Penroseovog inverza polinomijalnih matrica

Leverrier-Faddeev metod (Algoritam 3.3.2), izložen i opisan u odeljku 3.3 primenljiv je na racionalne i polinomijalne matrice $A(s)$. Međuitim, konačni izraz za $A^\dagger(s)$ biće nedefinisan kada je s nula polinoma (racionalne funkcije) $a_k(s)$. Generalisani inverz za ove vrednosti argumenta s može da se izračuna nezavisno Algoritmom 3.3.2 ili nekim drugim metodom.

Izložićemo analizu složenosti Algoritma 3.3.2 kada je ulazna matrica polinomijalna. Zbog jednostavnosti označimo $A'(s) = A(s)A^T(s)$ i $d' = \deg A'(s)$. Pošto je $A(s) \in \mathbb{R}[s]^{n\times m}$, važi $A'(s) \in \mathbb{R}[s]^{n\times n}$. Prema tome u koraku 5 treba da pomnožimo dve matrice reda $n \times n$. Ovo množenje može da se izvrši u vremenu $\mathcal{O}(n^3)$ kada je A konstantna matrica, ali u slučaju polinomijalne matrice odgovarajuće vreme je $\mathcal{O}\left(n^3 \cdot d' \cdot \deg B_{j-1}(s)\right)$. Može da se dokaže matematičkom indukcijom da važi nejednakost $\deg B_j(s) \leq j \cdot d'$ za $j = 0, \ldots, n$ gde je jednakost dostižna. Složenost operacije množenja matrica u koraku 5 jednaka je $\mathcal{O}(n^3 \cdot j \cdot d'^2)$. Slično, može da se pokaže da je složenost koraka 6 i 7 jednaka $\mathcal{O}(n \cdot j \cdot d')$. Prema tome, vreme izvršenja tela petlje u koracima 4-8, u j-toj iteraciji je $\mathcal{O}(n^3 \cdot j \cdot d'^2)$ dok je ukupno vreme izvršenja svih iteracija petlje jednako

$$\mathcal{O}\left(\sum_{j=1}^{n} n^3 \cdot j \cdot d'^2\right) = \mathcal{O}(n^5 \cdot d'^2). \tag{4.3}$$

Sada može da se vidi da je složenost Algoritma 3.3.2 za polinomijalne matrice n puta veća nego u slučaju konstantnih matrica (odeljak 3.3.2). U praksi, ova složenost je manja od (4.3) (nemaju svi elementi matrica $B_j(s)$, $A_j(s)$ i $A'(s)$ maksimalni stepen), ali je i dalje velika.

Sledeća definicija umnogome pojednostavljuje notaciju.

Definicija 4.2.1. *Definišimo k^A, a_i^A i B_i^A, kao vrednosti k, a_i i B_i za $i = 0, \ldots, n$, izračunate Algoritmom 3.3.2 kada je ulazna matrica A konstantna, racionalna ili polinomijalna matrica. Takođe označimo $a^A = a_{k^A}^A$ i $B^A = B_{k^A-1}^A$.*

Napomenimo da za retke ulazne matrice $A(s)$ postoje modifikacije Algoritma 3.3.2. Većinu njih su dali Karampetakis [64, 65, 66, 69], Stanimirović [124] i takođe Stanimirović i Karampetakis [129]. Za razliku od tih metoda, interpolacioni metod izložen u ovom poglavlju namenjen je kako gustim, tako i retkim matricama.

4.2.2 Glavna teorema i interpolacioni algoritam

Glavna ideja interpolacionog metoda je izračunavanje vrednosti polinoma $a^{A(s)}$ i $B^{A(s)}$ primenjujući Algoritam 3.3.2 na konstantne matrice a zatim rekonstrukcija vrednosti $a^{A(s)}$ i $B^{A(s)}$ koristeći interpolaciju.

Dobro je poznato da postoji jedan i samo jedan polinom $f(s)$ stepena $q \leq n$ koji uzima date vrednosti vrednosti $f(s_0), f(s_1), \ldots, f(s_n)$ u datim tačkama $s_0, s_1, \ldots, s_n$. Ovaj polinom se naziva interpolacioni polinom q-tog stepena. Tri značajna metoda interpolacije su [117]:

(i) direktni pristup koristeći Vandermondeovu matricu

(ii) Newtonova interpolacija,

(iii) Lagrangeova interpolacija.

Za računanje generalisanih inverza polinomijalnih matrica (a takođe, i u mnogim drugim primenama) pogodno je koristiti Newtonov interpolacioni polinom [107].

U sledećoj teoremi razmatramo koliki je broj interpolacionih tačaka dovoljan za izračunvanje vrednosti $k^{A(s)}$, polinomijalne matrice $B^{A(s)}$ kao i polinoma $a^{A(s)}$.

Teorema 4.2.1. *Neka je $A(s) \in \mathbb{R}[s]^{n\times m}$, $A'(s) = A(s)A(s)^T$, $d' = \deg A'(s)$ i $\kappa = k^{A(s)}$. Sledeća tvrđenja su ispunjena:*

(a) $\kappa = \max\{k^{A(s')} \mid s' \in \mathbb{R}\}$.

(b) *Neka su s_i, $i = 0, \dots, n \cdot d'$ međusobno različiti brojevi. Tada važi*

$$\kappa = \max\{k^{A(s_i)} \mid i = 0, \dots, n \cdot d'\}.$$

(c) *Polinomi $B^{A(s)}$ i $a^{A(s)}$ mogu da se izračunaju koristeći skup vrednosti $B^{A(s_i)}$ i $a^{A(s_i)}$, $i = 0, \dots, k^{A(s)} \cdot d'$.*

Dokaz.

(a) Označimo $\kappa = k^{A(s)}$. Iz Algoritma 3.3.2 imamo da je $\kappa = \max\{k \mid a_k^{A(s)} \neq 0\}$. Za svako $s' \in \mathbb{R}$, iz Algoritma 3.3.2 važi $k^{A(s')} \leq n$, tako da je skup $\mathcal{K} = \{k^{A(s')} \mid s' \in \mathbb{R}\}$ ograničen i ima maksimum $k_0 = \max \mathcal{K} = k^{A(s_0)}$ za neko $s_0 \in \mathbb{R}$. Pokazaćemo da je $\kappa = k_0$.

Iz definicije broja κ važi $a_\kappa^{A(s)} \neq 0$, pa sledi da da postoji $s' \in \mathbb{R}$ takvo da je $a_\kappa^{A(s')} = a_\kappa^{A(s)}(s') \neq 0$. Odavde sledi $\kappa \leq k^{A(s')} \leq k^{A(s_0)} = k_0$.

Sa druge strane, iz definicije $\kappa = k^{A(s)}$ imamo $a_{\kappa+t}^{A(s)}(s) = 0$ za sve $t = 1, \dots, n - \kappa$. Prema tome, važi

$$a_{\kappa+t}^{A(s')} = a_{\kappa+t}^{A(s)}(s') = 0$$

za svako $s' \in \mathbb{R}$ i $t = 1, \dots, n - \kappa$. Odavde sledi $k^{A(s')} \leq \kappa, \forall s' \in \mathbb{R}$ i $k_0 = k^{A(s_0)} \leq \kappa$. Time smo dokazali da je $\kappa = k_0$.

(b) Neka su s_i, $i = 0, \dots, n \cdot d'$ međusobno različiti realni brojevi i $k' = \max\{k^{A(s_i)} \mid i = 0, \dots, n \cdot \deg A'(s)\}$. Pokazaćemo da je $k' = \kappa$.

Pretpostavimo da važi $a_\kappa^{A(s)}(s_i) = 0$ za svako $i = 0, \dots, n \cdot d'$. U saglasnosti sa Algoritmom 3.3.2, stepen polinoma $a_\kappa^{A(s)}(s)$ je ograničen sa $\kappa \cdot d'$. Pošto je $\kappa \cdot d' \leq n \cdot d'$, važi $a_\kappa^{A(s)}(s) = 0$, što je u kontadikciji sa definicijom broja κ. Prema tome važi

$$(\exists i_0 \leq n)(a_\kappa^{A(s_{i_0})} = a_\kappa^{A(s)}(s_{i_0}) \neq 0),$$

odakle sledi da je $\kappa \leq k^{A(s_{i_0})} \leq k'$.

Sa druge strane, iz definicije κ važi $a_{\kappa+t}^{A(s)}(s) = 0$ za svako $t = 1, \dots n - \kappa$. Pošto je jednakost $a_{\kappa+t}^{A(s_i)} = a_{\kappa+t}^{A(s)}(s_i) = 0$ zadovoljena za svako $i = 0, \dots, n \cdot d'$, može da se zaključi da je $a_{\kappa+t}^{A(s_i)} = 0$. Prema tome $k^{A(s_i)} \leq \kappa$ važi za svako $i = 0, \dots, n \cdot d'$ pa dobijamo $k' \leq \kappa$. Ovim je kompletiran deo **(b)** dokaza.

(c) Lako može da se dokaže da vrednosti $B^{A(s)}(s_i)$ i $a^{A(s)}(s_i)$ mogu da se izraćunaju koristeći sledeće relacije:

$$B^{A(s)}(s_i) = B_{\kappa-1}^{A(s_i)} = \begin{cases} A'(s_i)^{\kappa - k^{A(s_i)} - 1} \left(A'(s_i) B^{A(s_i)} + a^{A(s_i)} I_n \right), & \kappa > \kappa_i \\ B^{A(s_i)}, & \kappa = \kappa_i \end{cases}$$

$$a^{A(s)}(s_i) = \begin{cases} a^{A(s_i)}, & k^{A(s_i)} = \kappa, \\ 0, & k^{A(s_i)} < \kappa. \end{cases}$$

Sada znamo vrednosti polinoma $B^{A(s)}$ i $a^{A(s)}$ u $\kappa \cdot d' + 1$ različitih tačaka. Iz $\deg B^{A(s)} \leq (\kappa - 1) \cdot d'$ i $\mathrm{dg} a^{A(s)} \leq \kappa \cdot d'$ važi da polinomi $B^{A(s)}$ i $a^{A(s)}$ mogu da se rekonstruišu iz skupa tačaka $B^{A(s)}(s_i)$ i $a^{A(s)}(s_i)$ $(i = 0, \dots, \kappa \cdot d')$ koristeći interpolaciju. $\square$

Na osnovu Teoreme 4.2.1 postavljamo sledeći interpolacioni algoritam. Prvo biramo različite realne brojeve s_i, $i = 0, \dots, n \cdot d'$, a zatim nalazimo $\kappa = k^{A(s)}$ koristeći izraz u delu **(b)** Teoreme 4.2.1. Nakon toga računamo vrednosti $B_{\kappa-1}^{A(s_i)}$ i $a_{\kappa}^{A(s_i)}$ za $i = 0, \dots, \kappa \cdot d'$ primenjujući Algoritam 3.3.2. Na kraju, na osnovu dobijenih vrednosti interpoliramo polinome $B_{\kappa-1}^{A(s)}$ i $a_{\kappa}^{A(s)}$.

Ova procedura je realizovana u Algoritmu 4.2.1.

Algoritam 4.2.1 Interpolacioni algoritam za izračunavanje Moore-Penroseovog inverza

Input: Polinomijalna matrica $A(s) \in \mathbb{R}[s]^{n \times m}$

1: $A'(s) := A(s)A(s)^T$
2: $d' := \deg A'(s)$
3: Odrediti međusobno različite brojeve $s_0, s_1, \dots, s_{nd'} \in \mathbb{R}$
4: **for** $i := 0$ to $n \cdot d'$ **do**
5: Primeni Algoritam 3.3.2 na ulaznu matricu $A(s_i)$, bez izvršavanja koraka **return** (korak 10).
6: $\kappa_i := k^{A_i}$
7: $B'_i := B_{\kappa_i - 1}^{A_i}$
8: $a'_i := a_{\kappa_i}^{A_i}$
9: **end for**
10: $\kappa := \max\{\kappa_i \mid i = 0, \dots, d\}$
11: **if** $\kappa = 0$ **then**
12: **return** $A^\dagger(s) := \mathbb{O}$
13: **else**
14: **for** $i := 0$ to $\kappa \cdot d'$ **do**
15: $A'_i := A'(s_i)$
16: $B_i := \begin{cases} {A'_i}^{\kappa-\kappa_i-1} \left(A'_i B'_i + a'_i I_n \right), & \kappa > \kappa_i \\ B'_i, & \kappa = \kappa_i \end{cases}$
17: $a_i := \begin{cases} 0, & \kappa > \kappa_i \\ a'_i, & \kappa = \kappa_i \end{cases}$
18: **end for**
19: Interpolirati polinom $a_{\kappa}^{A(s)}$ i polinomijalnu matricu $B_{\kappa-1}^{A(s)}$ koristeći parove (s_i, a_i) i (s_i, B_i), $i = 0, \dots, \kappa \cdot d'$ kao interpolacione tačke.
20: **return** $A^\dagger(s) := -\frac{1}{a_{\kappa}^{A(s)}(s)} A(s)^T B_{\kappa-1}^{A(s)}(s)$
21: **end if**

Napomenimo da u koracima 15-18 ažuriramo samo prvih $\kappa \cdot d'$ matrica B_i i brojeva a_i, jer je to dovoljno za korak 24. Ovo važi zbog činjenice da je $\deg B_{\kappa-1}^{A(s)}(s) \leq (\kappa - 1)d'$. Sledeća teorema pokazuje način na koji je moguće preračunati vrednosti $\kappa = k^{A(s)}$. Rezultat takvog preračunavanja je manje vreme izvršenja petlje u koracima 5-10. Broj tačaka potreban za interpolaciju $a^{A(s)}$ i $B^{A(s)}$ je $(\kappa - 1)d'$ i zato ova petlja ima granice 0 i $(\kappa - 1)d'$.

Sledeća teorema daje drugi način za izračunavanje vrednosti k^A. Dokaz ove teoreme nećemo izvoditi zato što se izvodi potpuno isto kao dokaz Teoreme 4.4.3, uz korišćenje činjenice da je $\mathrm{rank}A = \mathrm{rank}A'$.

Teorema 4.2.2. *Neka je* $A \in \mathbb{R}^{n \times m}$. *Tada važi* $\mathrm{rank}A = k^A$.

Koristeći Teoremu 4.2.2 i deo **(b)** Teoreme 4.2.1, možemo najpre da preračunamo vrednosti $\kappa = \max\{\text{rank} A'_i \,|\, i = 0, \ldots, n \cdot d'\}$ ($A'_i = A'(s_i)$), i posle toga u koracima 5-10 da odredimo $a_i = a_\kappa^{A(s_i)}$ i $B_i = B_{\kappa-1}^{A(s_i)}$ samo za $i = 0, \ldots, (\kappa - 1) \cdot d'$. Iako (kao što sledi iz analize složenosti date u nastavku) ovo poboljšanje ne smanjuje ukupnu složenost Algoritma 3.3.2, ono je veoma korisno u praksi. Ova modifikacija je iskorišćena u implementaciji Algoritma 4.2.1.

Izvršimo sada analizu složenosti Algoritma 4.2.1. U koracima 5-10 imamo petlju od $n \cdot d' + 1$ ciklusa. U svakom ciklusu računamo vrednosti a_i, B_i i κ_i koristeći Algoritam 3.3.2 za konstantnu matricu A_i. Složenost Algoritma 3.3.2 za konstantne matrice je $\mathcal{O}(n^4)$. Prema tome, složenost petlje u koracima 5-10 je $\mathcal{O}(n^5 \cdot d')$, ($d' = \deg A'(s)$). Petlja u koracima 15-18 (izračunavanje matrica B_i) se izvršava u vremenu $\mathcal{O}(n \cdot n^3 \cdot \log(\kappa - \kappa_i)) = \mathcal{O}(n^4 \cdot \log(n \cdot d'))$, koje je manje nego složenost petlje u koracima 5-10. Pretpostavljamo da se stepen matrica računa u vremenu $\mathcal{O}(\log(m))$ koristeći rekurzivne formule $A^{2l} = (A^l)^2$ i $A^{2l+1} = (A^l)^2 A$. Na kraju, složenost poslednjeg koraka (interpolacija) je $\mathcal{O}(n^4 \cdot d'^2)$ kada se koristi Newtonov interpolacioni metod. Prema tome složenost celog algoritma je $\mathcal{O}(n^4 \cdot d'^2 + n^5 \cdot d')$.

Ovako određena složenost je bolja (ali ne mnogo) od složenosti Algoritma 4.2.1 za polinomijalne matrice. Ali, kao što će biti pokazano u poslednjem odeljku, u praksi je Algoritam 4.2.1 mnogo bolji od Algoritma 3.3.2 posebno za guste matrice. Takođe, oba algoritma obično ne dostižu svoju maksimalnu složenost, što će isto tako biti pokazano u poslednjem odeljku.

4.2.3 Procena stepena polinoma $B_i^{A(s)}, a_i^{A(s)}$

U prethodnom odeljku pomenuta je nejednakost $\deg B_j^{A(s)} \leq j \cdot d'$. Tu i slične relacije koristili smo za analizu složenosti. U praksi, ova granica je retko dostižna jer samo nekoliko elemenata matrice $A'(s)$ (i drugih matrica) ima maksimalan stepen (jednak d').

Koristeći Lemu 4.1.1, možemo konstruisati sledeći algoritam za procenu gornje granice D_i^B i D_i^A matrica stepena $\text{dg} B_i^{A(s)}$ i $\text{dg} A_i^{A(s)}$ respektivno, kao i gornje granice d_i stepena polinoma $a_i(s)$.

Prema tome, potreban broj interpolacionih tačaka za rekonstrukciju polinomijalne matrice $(B_t^{A(s)})_{ij}$ jednak je $(D_t^B)_{ij}$, dok je za rekonstrukciju polinoma $a_t^{A(s)}$ potreban broj tačaka jednak d_t.

4.2.4 Implementacija

Algoritmi 3.3.2, 4.2.1 i 4.2.2 su implementirani u simboličkom programskom jeziku MATHEMATICA.

Funkcija `GeneralInv[A, p]` implementira blago modifikovanu verziju Algoritma 3.3.2.

```
GeneralInv[A_, p_] :=
    Module[{e, n, m, t, l, h, a, A1, B, k, at, Btm1, Btm2, AA, ID},
      R = A; T = A; e = 1;
      {n, m} = Dimensions[A];
      ID = IdentityMatrix[n];
      AA = Expand[A.Transpose[A]];
```

Algoritam 4.2.2 Procena stepena matrica $\mathrm{dg}B_t^{A(s)}(s)$ i stepena polinoma $\mathrm{dg}\left(a_t^{A(s)}\right)$ za datu matricu $A(s)$, $0 \le t \le n \cdot d'$.

Input: Matrica $A(s) \in \mathbb{R}^{n\times m}[s]$

$A'(s) = A(s)A(s)^T$

Stavi $(D_0^B)_{ii} := 0$ za svako $i = 1, \dots, n$

Stavi $(D_0^B)_{ij} := -\infty$ za svako $i = 1, \dots, n$, $j = 1, \dots, n$, $i \neq j$

$Q := \mathrm{dg}A'(s)$

$d_0 := 0$

for $t = 1$ to n **do**

  $(D_t^A)_{ij} := \max\{Q_{ik} + (D_{t-1}^B)_{kj} \,|\, k = 1, \dots, n\}$, za $i = 1, \dots, n$, $j = 1, \dots, n$

  $d_t := \max\{(D_t^A)_{ii} \,|\, i = 1, \dots, n\}$

  $(D_t^B)_{ii} := \max\{(D_t^A)_{ii}, d_t\}$ za svako $i = 1, \dots, n$

  $(D_t^B)_{ij} := (D_t^A)_{ij}$ za svako $i = 1, \dots n$, $j = 1, \dots n$, $i \neq j$.

end for

return $\{D_t^B\}_{0\le t\le n}$ i $\{d_t\}_{0\le t\le n}$

```
      B = IdentityMatrix[n]; t = -1; l = -1; a = 1;
      Btm1 = B;
      For [h = 1, h <= n, h++,
        A1 = Expand[AA.B];
        a = Expand[-1/h*Tr[A1]];
        If [a =!= 0, t = h; at = a; Btm2 = B;];
        Btm1 = B;
        B = Expand[A1 + a*ID];
        If [h == p, Return[{t, Expand[a], Expand[Btm1]}];];
        ];

      Return[{t, Expand[at], Expand[Btm2]}];
      ];
```

Za ulaznu matricu $A \in \mathbb{R}^{n\times m}$ i pozitivan ceo broj p, ova funkcija vraća listu sa elementima p, a_p^A i B_{p-1}^A respektivno, ako je $0 \le p \le n \cdot d'$. U suprotnom, funkcija vraća listu sa elementima k^A, $a_{k^A}^A$ i $B_{k^A-1}^A$. Funkcija radi za racionalne, polinomijalne i konstantne matrice.

Funkcija `DegreeEstimator[A, i, var]` implementira Algoritam 4.2.2 i daje gornju granicu za stepen polinoma $a_i^{A(s)}$ i matricu stepena $\mathrm{dg}B_{i-1}^{A(s)}$.

```
DegreeEstimator[A_, i_, var_] :=
    Module[{h, j, k, l, m, d1, d2, Bd, ad, AA, Ad, Btm1d, Btm2d, atd, td,
        IDd},
      {d1, d2} = Dimensions[A];
      Ad = MatrixDg[A, var];
      Ad = MultiplyDG[Ad, Transpose[Ad]];
      Bd = MatrixDg[IdentityMatrix[d1], var];

      IDd = Bd; td = -1; l = -1; ad = -\[Infinity];
      For [h = 1, h <= i, h++,

        A1d = MultiplyDG[Ad, Bd];
```

```
        ad = Max[Table[A1d[[j, j]], {j, d1}]];

        td = h; atd = ad; Btm2d = Bd;

        Btm1d = Bd;
        Bd = A1d;
        For [j = 1, j <= d1, j++,
          Bd[[j, j]] = Max[Bd[[j, j]], ad];
          ];
        ];
      Return[{atd, Btm2d}];
      ];
```

Glavna petlja funkcije `DegreeEstimator` (u odnosu na promenljive `h`) implementira korake 6-11 Algoritma 4.2.2. Ova funkcija koristi dve pomoćne funkcije:

- Funkcija `MatrixDg[A, var]` izračunava matricu stepena ulazne matrice `A`,
- Funkcija `MultiplyDG[Ad,Bd]` izračunava gornju granicu matrice stepena za proizvod matrica A i B čije su matrice stepena jednake `Ad` i `Bd`.

Obe funkcije su zasnovane na Lemi 4.1.1. Funkcija `GeneralInvPoly[A, var]` implementira blagu modifikaciju Algoritma 4.2.1.

```
GeneralInvPoly[A_, var_] :=
    Module[{dg, tg, deg, n, m, x, tm, i, h, p, Ta, TB, A1, a, B, t, at, Btm1,
        r1, r, degA, Deg},

        {n, m} = Dimensions[A];
        degA = MatrixPolyDegree[A, s];
        p = n*2*degA + 1;

        x = Table[i, {i, 1, p}];
        r = 0;

        AA = Expand[A.Transpose[A]];
        For [h = 1, h <= p, h++,
          r1 = MatrixRank[ReplaceAll[AA, var -> x[[h]]]];
          If [r1 > r, r = r1];
        ];

        tm = -1; tg = 0; p = r*2*degA + 1;
        Ta = Table[0, {i, 1, p}]; TB = Table[0, {i, 1, p}];
        For [h = 1, h <= p, h++,
          A1 = ReplaceAll[A, var -> x[[h]]];
          {t, a, B} = GeneralInv[A1, r];
          Ta[[h]] = {h, a}; TB[[h]] = {h, B};
        ];

       {deg, Deg} = DegreeEstimator[A, r, var];
       at = SimpleInterpolation[Ta, deg, var];
       Btm1 = AdvMatrixMinInterpolation[TB, Deg, var];
```

```
    Return[{Expand[at], Expand[Btm1]}];
];
```

Ulazni parametar ove funkcije je polinomijalna matrica $A(s)$ (promenljiva je u kodu označena sa `var`). Prva i druga dimenzija matrice A su jednake n i m, respektivno. Funkcija vraća listu elemenata $\kappa = k^{A(s)}$, $a_\kappa^{A(s)}$ i $B_{\kappa-1}^{A(s)}$. U ovoj implementaciji koristili smo $s_i = i$ kao interpolacione tačke. Sa ovim skupom interpolacionih tačaka, vreme izvršenja funkcije je najmanje (takođe smo probali $s_i = -[\frac{n}{2}] + i$, $s_i = \frac{i}{n}$, itd.). Prva petlja (u odnosu na promenljivu `h`) izračunava rang svake od matrica $A_i = A(s_i)$ (označen sa `r1`) kao i maksimum κ ovih rangova (označenih sa `r`). Druga petlja izračunava $a_i = a_\kappa^{A(s_i)}$ i $B_i = B_{\kappa-1}^{A(s_i)}$, koristeći funkciju `GeneralInv[A, p]` za konstantnu matricu $A(s_i)$ (označenu u kodu sa `A1`). Izračunate vrednosti se čuvaju u listama `Ta` i `TB`. Posle toga, procenjujemo stepen polinoma $a_\kappa^{A(s)}$ kao i matricu stepena polinomijalne matrice $B_{\kappa-1}^{A(s)}$ koristeći funkciju `DegreeEstimator[A, i, var]`.

Napomenimo na kraju da unutar funkcije `GeneralInvPoly` koristimo pomoćne funkcije

- `SimpleInterpolation[Ta, deg, var]`,
- `AdvMatrixMinInterpolation[TB, Deg, var]`,

koje interpoliraju polinome $a_\kappa^{A(s)}$ i $B_{\kappa-1}^{A(s)}$ kroz izračunate tačke. Obe funkcije koriste funkciju `InterpolatingPolynomial[T, var]` ugrađenu u programski paket MATHEMATICA , zasnovanu na Newtonovom interpolacionom metodu.

Napomena 4.2.1. *Kada imamo fiksiran maksimalni stepen polinomijalne matrice koju želimo rekonstruisati interpolacijom, možemo da koristimo Lagrangeov metod sa preračunatim koeficijentima*

$$L_i(s) = \frac{\prod_{j=0,j\neq i}^{n}(s - s_j)}{\prod_{j=0,j\neq i}^{n}(s_i - s_j)}.$$

Dalje možemo da izračunamo interpolacioni polinom kao $f(s) = \sum_{i=0}^{d} f_i \cdot L_i(s)$, gde je $f(s)$ bilo $a^{A(s)}$ ili $B^{A(s)}$. Vremenska složenost ove procedure je $\mathcal{O}(d^2)$, kao i u slučaju Newtonovog metoda, gde je d stepen interpolacionog polinoma. U našem slučaju, ovaj metod je bio nekoliko puta sporiji jer je ugrađena funkcija `InterpolatingPolynomial[T, var]` *mnogo brža od naše funkcije implementirane u programskom paketu* MATHEMATICA .

4.2.5 Rezultati testiranja

Testiraćemo implementacije Algoritma 3.3.2 i Algoritma 4.2.1 poboljšanog Algoritmom 4.2.2 na test matricama iz [162] i nekim slučajno generisanim test matricama. U sledećoj tabeli prikazano je vreme izvršenja funkcija `GeneralInv` i `GeneralInvPoly` na test primerima iz [162]. Sva vremena su u sekundama.

Matrica	Alg 3.3.2	Alg 4.2.1
S_3	0.047	0.078
S_6	0.391	0.547
S_{10}	2.015	5.469
V_4	0.078	1.219
V_5	0.593	17.11
H_3	0.015	0.032
H_6	0.047	0.187
H_{10}	0.5	2.109

Ove matrice su veoma retke, tako da je Algoritam 4.2.1 (funkcija `GeneralInvPoly`) sporiji od Algoritma 3.3.2 (funkcija `GeneralInv`).

Primer 4.2.1. Razmotrimo test matricu $V_n(a,b)$ definisanu sledećom rekurentnom formulom [162]

$$V_0(a,b) = \begin{pmatrix} a & b \\ b & -a \end{pmatrix}, \quad V_n(a,b) = \begin{pmatrix} V_{n-1}(a,b) & V_{n-1}(a,b) \\ V_{n-1}(a,b) & -V_{n-1}(a,b) \end{pmatrix}.$$

Označimo $V_n(s) = V_n(s,s)$. Važe sledeće relacije

$$k^{V_n(s)} = 2^n - 1, \quad a^{V_n(s)} = n2^n \cdot s^{2^{n+1}},$$
$$B^{V_n(s)} = -2^{n(2^n-1)} s^{2^n-1} \cdot I_{2^n}, \quad V_n^{\dagger}(s) = V_n\left(\frac{1}{2^n s}\right).$$

Kao što možemo da vidimo, samo elementi glavne dijagonale matrice $B^{V_n(s)}$ su različiti od nule i imaju tačno jedan sabirak (samo prvi koeficijent nije nula). Slično važi za sve matrice $B_i^{V_n(s)}$, $0 \le i \le 2^n - 1$. Ovo objašnjava loš rezultat Algoritma 4.2.1 na test matrici $V_5(s)$. Sličan zaključak se može izvući i za ostale test matrice iz prethodne tabele.

Kao što će biti pokazano, spars brojevi $sp_1(A)$ i $sp_2(A)$ imaju uticaj na vreme izvršenja Algoritma 3.3.2 i Algoritma 4.2.1. Naša funkcija `RandomMatrix[n, deg, prob1, prob2, var]` slučajno generiše polinomijalnu matricu A formata $n \times n$ u odnosu na promenljivu `var` čiji je stepen ($\deg A$) jednak `deg` a dva spars broja ($sp_1(A)$ i $sp_2(A)$) su jednaka `prob1` i `prob2`. U sledećim tabelama prikazano je srednje vreme izvršenja (srednja vrednost vremena izvršenja za 10 različitih slučajno generisanih test matrica istog tipa) oba algoritma za $n = 9, 10, 11$ i $\deg A = 3, 4, 5$.

n	$\deg A$	Alg 3.3.2	Alg 4.2.1
9	3	3.6376	1.4844
9	4	6.1846	2.3528
9	5	8.6374	3.4846
10	3	6.2002	2.275
10	4	10.0246	3.6346
10	5	14.8842	5.425
11	3	10.1686	3.3718
11	4	16.5126	5.4312
11	5	24.603	8.116

$sp_1(A) = sp_2(A) = 1$

n	$\deg A$	Alg 3.3.2	Alg 4.2.1
9	3	2.2064	1.1594
9	4	4.35	2.1096
9	5	6.3814	2.928
10	3	4.7216	2.0752
10	4	6.6312	3.072
10	5	11.6094	4.878
11	3	7.8282	3.05
11	4	12.797	4.8532
11	5	19.869	7.45

$sp_1(A) = sp_2(A) = 0.5$

n	$\deg A$	Alg 3.3.2	Alg 4.2.1
9	3	0.8748	0.8126
9	4	1.5968	1.444
9	5	1.4998	1.6374
10	3	1.6032	1.225
10	4	3.2438	2.2688
10	5	4.8218	3.497
11	3	3.4502	2.481
11	4	6.4814	4.1218
11	5	10.4158	6.862

$sp_1(A) = sp_2(A) = 0.35$

Za guste matrice ($sp_1(A) = sp_2(A) = 1$), Algoritam 4.2.1 je mnogo brži u odnosu na Algoritam 3.3.2 na svim test primerima. Za $sp_1(A) = sp_2(A) = 0.5$ Algoritam 3.3.2 je malo brži ali još uvek sporiji nego Algoritam 4.2.1. Napomenimo da matrice $B_i^{A(s)}(s)$ najčešće imaju veće spars brojeve (osnovne matrične operacije obično povećavaju spars brojeve), tako da je situacija slična kao i u prvom slučaju. U trećem slučaju ($sp_1(A) = sp_2(A) = 0.35$), algoritmi su skoro jednako brzi (Algoritam 4.2.1 je samo malo brži u većini test primera). Za manje spars brojeve Algoritam 3.3.2 je brži nego Algoritam 4.2.1. Napomenimo da, kada spars brojevi opadaju (uz uslov $sp = sp_1(A) = sp_2(A)$) vreme izvršenja Algoritma 3.3.2 opada naglo, što nije slučaj sa Algoritmom 4.2.1, čije vreme izvršenja opada sporo. Ovo objašnjava činjenica da vreme izvršenja polinomijalnih operacija u Algoritmu 3.3.2 zavisi od broja nenula koeficijenata, koji opada linearno sa sp (dok je $\deg A$ konstantno). U Algoritmu 4.2.1, vreme izvršenja interpolacije zavisi od stepena polinoma koji se interpoliraju, a on opada sporije sa sp.

Razmotrimo sada dva ekstremna slučaja, prikazana u sledećim tabelama.

n	$\deg A$	Alg 3.3.2	Alg 4.2.1
9	3	1.8842	1.0532
9	4	3.1	1.7498
9	5	4.4876	2.5748
10	3	3.2688	1.6282
10	4	5.5688	2.7936
10	5	7.5816	3.8094
11	3	5.7592	2.4592
11	4	8.8656	3.9622
11	5	15.2844	7.0502

$sp_1(A) = 1$, $sp_2(A) = 0.1$

n	$\deg A$	Alg 3.3.2	Alg 4.2.1
9	3	0.1188	0.2968
9	4	0.1032	0.2656
9	5	0.2154	0.4062
10	3	0.1908	0.4092
10	4	0.331	0.5876
10	5	0.4468	0.7814
11	3	0.5624	0.6752
11	4	0.5594	0.8874
11	5	0.7158	1.0718

$sp_1(A) = 0.1$, $sp_2(A) = 1$

U prvoj tabeli, Algoritam 3.3.2 je sporiji od Algoritma 4.2.1. Kad se n i $\deg A$ povećavaju, razlika (odnos) između tih vremena se povećava. Napomenimo da je vreme izvršenja Algoritma 4.2.1 u ovom slučaju i u slučaju $sp_1(A) = sp_2(A) = 1$ skoro isto. Objašnjenje ove činjenice je slično kao i u prethodnom slučaju. U drugom slučaju, Algoritam 3.3.2 je brži, i vreme izvršenja je značajno manje nego u prethodnim slučajevima. Ovde imamo mali ukupan broj nenula koeficijenata u svim elementima matrice A.

4.3 Izračunavanje Drazinovog inverza polinomijalnih matrica interpolacijom

U ovom poglavlju konstruisaćemo interpolacioni metod za izračunavanje Drazinovog inverza date polinomijalne matrice. Ovaj metod je zasnovan na Leverrier-Faddeevom metodu za izračunavanje Drazinovog inverza, koji je uveden u odeljku 3.3.3. Interpolacioni metod je poboljšan korišćenjem metoda za procenu matrica stepena. Algoritmi su implementirani i testirani u simboličkom paketu MATHEMATICA.

Rezultati sadržani u ovom poglavlju su originalni i preuzeti iz našeg rada [103]. Kao i u prethodnom, i u ovom poglavlju bavićemo se polinomijalnim matricama sa **realnim** koeficijentima.

4.3.1 Leverrier-Faddeev metod za izračunavanje Drazinovog inverza polinomijalnih matrica

U ovom odeljku, biće prikazana analiza složenosti Algoritma 3.3.3 za polinomijalne matrice. Radi jednostavnije notacije, redefinisaćemo vrednosti k^A, a_i^A i B_i^A date Definicijom 4.2.1.

Definicija 4.3.1. *Označimo sa k^A, t^A, a_i^A i B_i^A vrednosti k, t, a_i i B_i, respektivno, u Algoritmu 3.3.3 kada je njegov ulaz polinomijalna ili konstantna matrica A. Takođe, koristićemo jednostavnije oznake $a^A = a^A_{k^A}$ i $B^A = B^A_{k^A-1}$.*

Do kraja ovog poglavlja pretpostavićemo da su ove vrednosti određene saglasno Definiciji 4.3.1. Sledeća lema može lako da se dokaže matematičkom indukcijom, i biće veoma korisna u daljim razmatranjima.

Lema 4.3.1. *Neka je $A \in \mathbb{C}^{n\times n}$ konstantna, racionalna ili polinomijalna matrica. Tada važi*

(a) $B^A_{t^A+i} = 0$ *za svako* $i = 0, \ldots, n - t^A - 1$,

(b) $B^A_{k^A+i-1} = A^{i-1}\left(AB^A + a^A I_n\right)$ *za svako* $i = 1, \ldots, t^A - k^A$,

(c) $a^A_{t^A+i} = 0$ *za svako* $i = 0, \ldots, n - t^A$ *ili ekvivalentno* $k^A \le t^A$.

(d) *Ako je* $A = A(s) \in \mathbb{R}[s]^{n\times n}$ *tada važi* $\deg B_i(s) \le i \cdot \deg A(s)$ *i* $\mathrm{dg} a_i(s) \le i \cdot \deg A(s)$ *za svako* $i = 0, \ldots, n$.

Sada ćemo pristupiti analizi služenosti Algoritma 3.3.3 kada je ulazna matrica polinomijalna $A(s) \in \mathbb{R}[s]^{n\times n}$.

U koraku 5 treba da pomnožimo dve matrice reda $n \times n$. Ovo množenje se vrši u vremenu $\mathcal{O}(n^3)$ kada je A konstantna matrica, ali kada je A polinomijalna matrica, odgovarajuća složenost je $\mathcal{O}\left(n^3 \cdot \deg A(s) \cdot \deg B_{j-1}(s)\right)$. Ako označimo $d = \deg A(s)$, tada u saglasnosti sa delom **(d)** Leme 4.3.1, vreme potrebno za izvršenje koraka 5 je $\mathcal{O}(n^3 \cdot j \cdot d^2)$. Slično, može da se pokaže da je vreme potrebno za korak 6 kao i za korak 7 jednako $\mathcal{O}(n \cdot j \cdot d)$. Ovo je znatno manje od vremena izvršenja koraka 5, tako da je ukupno vreme izvršenja koraka 5-7, u j-tom ciklusu petlje jednako $\mathcal{O}(n^3 \cdot j \cdot d^2)$. Ukupna složenost **for** petlje u koracima 4-8 je

$$\mathcal{O}\left(\sum_{j=1}^{n} n^3 \cdot j \cdot d^2\right) = \mathcal{O}(n^5 \cdot d^2) \tag{4.4}$$

Razmotrimo sada složenost ostatka Algoritma 3.3.3. Vreme izvršenja koraka 9 je $\mathcal{O}(n^2 \cdot d)$, dok je za korak 10 to vreme jednako

$$\mathcal{O}\left(n^2 \cdot \sum_{j=1}^{n} j \cdot d\right) = \mathcal{O}(n^4 \cdot d).$$

U koraku 12 treba naći k-ti i $k+1$-vi stepen matrica $A(s)$ i $B_{t-1}(s)$ respektivno. Složenost ovog koraka je manja od (4.4). Prema tome, ukupno vreme izvršenja (složenost) Algoritma 3.3.3 kada je primenjen na polinomijalnu matricu je $\mathcal{O}(n^5 \cdot d^2)$. U praksi, složenost Algoritma 3.3.3 je manja od (4.4) (nemaju svi elementi matrica $B_j(s)$, $A_j(s)$ i $A(s)$ maksimalan stepen).

Za retke ulazne matrice $A(s)$, modifikacija Algoritma 3.3.3 data je u radu Stanimirovića i Tasića [134]. Umesto ovih metoda, interpolacioni metod, izložen u ovom poglavlju je pogodan kako za guste tako i za retke matrice.

4.3.2 Glavna teorema i interpolacioni algoritam

Slično kao u predhodnom poglavlju i ovde dajemo teoremu koja određuje dovoljan broj interpolacionih tačaka za izračunavanje vrednosti $k^{A(s)}$, $t^{A(s)}$ i rekonstrukciju polinoma $B^{A(s)}, a^{A(s)}$.

Teorema 4.3.2. *Neka je $A(s) \in \mathbb{R}[s]^{n\times n}$, $d = \deg A(s)$, $\tau = t^{A(s)}$ i $\kappa = k^{A(s)}$. Neka su s_i, $i = 0, \dots, n\cdot d$ međusobno različiti realni brojevi. Tada važe sledeća tvrđenja:*

(a) *Ako označimo $f(j) = \max\{t^{A(s_i)} \mid i = 0, \dots, j\cdot d\}$ za $j = 1, \dots, n$, tada je τ je jedinstven broj koji zadovoljava $\tau = f(\tau)$, i takođe važi $\tau = f(n)$.*

(b) $\kappa = \max\{k^{A(s_i)} \mid i = 0, \dots, \tau \cdot d\}$.

(c) *Polinomijalna matrica $B^{A(s)}$ i polinom $a^{A(s)}$ mogu da se izračunaju koristeći skup konstantnih matrica $B^{A(s_i)}$ i vrednosti $a^{A(s_i)}$ respektivno, za $i = 0, \dots, \kappa\cdot d$.*

Dokaz.

(a) Iz Leme 4.3.1 imamo $B_i^{A(s)} = 0 \iff i \geq t^{A(s)}$.

Najpre ćemo dokazati da je $f(j) \leq \tau$ za svako $j = 1, \dots, n$. Pretpostavimo da je $f(j_0) > \tau$ za neko $j_0 \in \{1, \dots, n\}$. Pošto postoji $0 \leq i_0 \leq d$ takvo da je $t^{A(s_{i_0})} = f(j_0)$ i pošto je $t^{A(s_{i_0})} > \tau$, važi $B_\tau^{A(s)}(s_{i_0}) = B_\tau^{A(s_{i_0})} \neq 0$. Odavde sledi $B_\tau^{A(s)} \neq 0$, što je kontradikcija. Pošto je $0 \leq \tau \leq n$, imamo $f(\tau) \leq \tau$.

Dokažimo sada suprotnu nejednakost. Pretpostavimo da je $t' = f(\tau) < \tau$. Tada iz $t' \geq t^{A(s_i)}$ imamo $B_{t'}^{A(s)}(s_i) = B_{t'}^{A(s_i)} = 0$ za $i = 0, \dots, t'\cdot d$. Iz Leme 4.3.1 imamo $\deg B_{t'}^{A(s)} \leq t'\cdot d < t'\cdot d+1$, tako da možemo da zaključimo da je $B_{t'}^{A(s)} = 0$, što je kontradikcija sa definicijom broja τ. Prema tome, $f(\tau) = \tau$.

Dalje u ovom delu dokaza, pokazaćemo da f ima jedinstvenu fiksnu tačku. Drugim rečima potrebno je dokazati da ako ako za bilo koje $1 \le t_0 \le n$ važi $t_0 = f(t_0)$, onda je $t_0 = \tau$. Iz definicije t_0 imamo da je $t_0 \ge t^{A(s_i)}$ za svako $i = 0, \dots, t_0 \cdot d$. Takođe iz definicije $t^{A(s_i)}$, i iz Leme 4.3.1 imamo $B_{t_0}^{A(s)}(s_i) = B_{t_0}^{A(s_i)} = 0$, $i = 0, \dots t_0 \cdot d$. Ponovo, iz Leme 4.3.1 dobijamo da važi $\deg B_{t_0}^{A(s)} \le t_0 \cdot d$, tako da možemo da zaključimo da je $B_{t_0}^{A(s)} = 0$. Ovim dokazujemo da je $\tau \le t_0$. Tvrđenje $t_0 = f(t_0) \le \tau$ smo već dokazali. Ovim smo kompletirali dokaz da je $t_0 = \tau$. Pošto je $f(j) \le \tau$, za svako $j = 1, \dots, n$, odmah zaključujemo da važi $\tau \ge f(n)$. Iz definicije f, imamo $f(i) \ge f(j)$ za $i > j$. Tada iz $\tau \ge f(n) \ge f(\tau) \ge \tau$ sledi $\tau = f(n) = f(\tau)$.

(b) Neka je $k' = \max\{k^{A(s_i)} | i = 0, \dots, \tau \cdot d\}$. Pokazaćemo da je $k' = \kappa$.

Pretpostavićemo da je $a_\kappa^{A(s)}(s_i) = 0$ za svako $i = 0, \dots, \tau \cdot d$. Saglasno Lemi 4.3.1, stepen polinoma $a_\kappa^{A(s)}(s)$ je ograničen sa $\kappa \cdot d$. Pošto je $\kappa \cdot d \le \tau \cdot d$, imamo $a_\kappa^{A(s)}(s) = 0$, što je u kontradikciji sa definicijom broja κ. Prema tome važi

$$(\exists i_0 \le \tau \cdot d)(a_\kappa^{A(s_{i_0})} = a_\kappa^{A(s)}(s_{i_0}) \ne 0),$$

odakle sledi $\kappa \le k^{A(s_{i_0})} \le k'$.

Sa druge strane, iz definicije κ imamo $a_{\kappa+j}^{A(s)}(s) = 0$ za svako $t = 1, \dots n - \kappa$. Pošto je jednakost $a_{\kappa+j}^{A(s_i)} = a_{\kappa+j}^{A(s)}(s_i) = 0$ zadovoljena za svako $i = 0, \dots, \tau \cdot d$, može da se zaključi da je $a_{\kappa+j}^{A(s_i)} = 0$. Ovo znači da je $k^{A(s_i)} \le \kappa$ za svako $i = 0, \dots, \tau \cdot d$, odnosno $k' \le \kappa$. Na taj način je kompletiran ovaj deo dokaza.

(c) Iz Leme 4.3.1, za svako $i = 0, \dots, \kappa \cdot d$ imamo

$$B^{A(s)}(s_i) = B_{\kappa-1}^{A(s_i)} = \begin{cases} A(s_i)^{\kappa - k^{A(s_i)} - 1} \left(A(s_i) B^{A(s_i)} + a^{A(s_i)} I_n \right), & \kappa > \kappa_i \\ B^{A(s_i)}, & \kappa = \kappa_i \end{cases}$$

i

$$a^{A(s)}(s_i) = \begin{cases} a^{A(s_i)}, & k^{A(s_i)} = \kappa, \\ 0, & k^{A(s_i)} < \kappa. \end{cases}$$

Sada znamo vrednosti polinoma $B^{A(s)}$ i $a^{A(s)}$ u $\kappa \cdot d + 1$ različitih tačaka. Posle još jedne primene Leme 4.3.1, dokazujemo i deo pod **(c)**. $\square$

Prethodna teorema daje glavnu ideju za sledeći interpolacioni algoritam. Prvo se biraju različiti realni brojevi s_i, $i = 0, \dots, n \cdot d$, i nalaze $\tau = t^{A(s)}$ i $\kappa = k^{A(s)}$ iz tvrđenja **(a)** i **(b)** Teoreme 4.3.2. Dalje se računaju vrednosti $B_{\kappa-1}^{A(s_i)}$ i $a_\kappa^{A(s_i)}$ za $i = 0, \dots, \kappa \cdot d$ koristeći deo **(c)** Leme 4.3.1, i računa polinomijalna matrica $B_{\kappa-1}^{A(s)}$ i polinom $a_\kappa^{A(s)}$ metodom interpolacije. Na kraju se računa Drazinov inverz koristeći korak 12 Algoritma 3.3.3.

Izvršavanjem prethodno navedenih koraka dolazimo do Algoritma 4.3.1.

Algoritam 4.3.1 Interpolacioni algoritam za izračunavanje Drazinovog inverza date polinomijalne matrice $A(s)$.

Input: Matrica $A(s) \in \mathbb{R}[s]^{n\times n}$

$d := \deg A(s)$

$d' := n \cdot d$

Odrediti različite realne brojeve $s_0, s_1, \cdots, s_{d'} \in \mathbb{R}$

$i := -1$

$\tau := -1$

repeat

  $i := i + 1$

  $A_i := A(s_i)$

  Primeniti Algoritam 3.3.3 na ulaznu matricu A_i, bez izvršavanja koraka **return** (korak 12).

  $\kappa_i := k^{A_i}$

  $\tau_i := t^{A_i}$

  $B_i' := B_{\kappa_i - 1}^{A_i}$

  $a_i' := a_{\kappa_i}^{A_i}$

  $\tau := \max\{\tau_i, \tau\}$

until $(i = \tau d)$ ili $(i = d')$

$\kappa := \max\{\kappa_i \mid i = 0, \ldots, \tau \cdot d\}$

if $\kappa = 0$ **then**

  return $A^D(s) := \mathbb{O}$

else

  for $i = 0$ to $\kappa \cdot d$ **do**

    $$B_i := \begin{cases} A_i^{\kappa-\kappa_i-1}\left(A_i B_i' + a_i' I_n\right), & \kappa > \kappa_i \\ B_i' & \kappa = \kappa_i \end{cases}$$

    $$a_i := \begin{cases} 0, & \kappa > \kappa_i \\ a_i', & \kappa = \kappa_i \end{cases}$$

  end for

  Interpolirati polinom $a_\kappa(s)$ i polinomijalnu matricu $B_{\kappa-1}(s)$ koristeći parove (s_i, a_i) i (s_i, B_i), $i = 0, \ldots, \kappa \cdot d$ kao interpolacione tačke.

  return $A^D(s) := (-1)^{r+1} a_\kappa(s)^{-r-1} A(s)^r B_{\kappa-1}(s)^{r+1}$

end if

Izvršićemo analizu složenosti Algoritma 4.3.1. Prvo, imamo petlju koja se ponavlja $\kappa \cdot d + 1$ puta (koraci 6-15). U svakom ciklusu izračunavamo vrednosti a_i, B_i, κ_i i τ_i koristeći Algoritam 3.3.3 za konstantne matrice A_i. Složenost Algoritma 3.3.3 za konstantne matrice je $\mathcal{O}(n^4)$. Prema tome, složenost petlje u koracima 6-15 je $\mathcal{O}(n^4 \cdot d') = \mathcal{O}(n^5 \cdot d)$ $(d = \deg A(s))$. Složenost petlje u koracima 20-23 je $\mathcal{O}(nd \cdot n^3 \cdot \log(\kappa - \kappa_i)) = \mathcal{O}(n^4 \cdot \log(n \cdot d))$, što je manje od složenosti petlje u koracima 6-15. Pretpostavljamo da se stepen matrice računa u vremenu $\mathcal{O}(\log(m))$ koristeći rekurzivne formule $A^{2l} = (A^l)^2$ i $A^{2l+1} = (A^l)^2 A$. Ove formule se takođe koriste u koraku 25. Na kraju, složenost koraka 24 (interpolacija) je $\mathcal{O}(n^2 \cdot d'^2) = \mathcal{O}(n^4 \cdot d^2)$, kada se koristi Newtonov interpolacioni metod. Prema tome, složenost celog algoritma je $\mathcal{O}(n^4 \cdot d^2 + n^5 \cdot d)$.

Dobijena složenost je bolja (ali ne mnogo) nego složenost Algoritma 3.3.3 za polinomijalne matrice. Ali kako što ćemo pokazati u zadnjem odeljku, u praksi je Algoritam 4.3.1 mnogo bolji nego Algoritam 3.3.3, posebno za guste matrice. Takođe treba napomenuti da oba algoritma obično ne dostižu svoju maksimalnu složenost.

Sličnim razmatranjem kao u odeljku 4.2.3, ovde takođe možemo da konstruišemo algoritam procene matrica stepena za polinomijalne matrice $B_i^{A(s)}$ i polinome $a_i^{A(s)}$, $i = 0, \dots, nd$.

Algoritam 4.3.2 Procena stepena matrica $\mathrm{dg} B_t^{A(s)}(s)$ i stepena polinoma $\mathrm{dg}\left(a_t^{A(s)}\right)$ za datu matricu $A(s)$, $0 \le t \le n \cdot d'$.

Input: Matrica $A(s) \in \mathbb{R}^{n \times n}[s]$

Stavi $(D_0^B)_{ii} := 0$ za svako all $i = 1, \dots, n$

Stavi $(D_0^B)_{ij} := -\infty$ za svako $i = 1, \dots, n$, $j = 1, \dots, n$, $i \ne j$

$Q := \mathrm{dg} A(s)$

$d_0 := 0$

for $t = 1$ to n **do**

    $(D_t^A)_{ij} := \max\{Q_{ik} + (D_{t-1}^B)_{kj} \mid k = 1, \dots, n\}$, za $i = 1, \dots, n$, $j = 1, \dots, n$

    $d_t := \max\{(D_t^A)_{ii} \mid i = 1, \dots, n\}$

    $(D_t^B)_{ii} := \max\{(D_t^A)_{ii}, d_t\}$ za svako $i = 1, \dots, n$

    $(D_t^B)_{ij} := (D_t^A)_{ij}$ za svako all $i = 1, \dots n$, $j = 1, \dots n$, $i \ne j$.

end for

return $\{D_t^B\}_{0 \le t \le n}$ i $\{d_t\}_{0 \le t \le n}$

Napomenimo još jednom, što je to uradjeno u odeljku 4.2.3, da je potreban broj interpolacionih tačaka za rekonstrukciju polinomijalne matrice $(B_l^{A(s)})_{ij}$ jednak $(D_l^B)_{ij}$ a za polinom $a_l^{A(s)}$ je d_l.

4.3.3 Implementacija

Algoritmi 3.3.3, 4.3.1 i 4.3.2 su implementirani u simboličkom programskom jeziku MATHEMATICA.

Funkcija `GeneralInvDrazin[A]` implementira Algoritam 3.3.3.

```
GeneralInvDrazin[A_] :=
    Module[{e, n, m, t, l, h, a, A1, B, k, at, Btm1, Btm2, AA, ID, av, Bv,
        vkk},
```

```
      {n, m} = Dimensions[A];
      ID = IdentityMatrix[n];
      B = IdentityMatrix[n]; k = 0; l = -1; a = 1; t = n;
      at = 0; Btm2 = 0*B;
      Btm1 = B;
      For [h = 1, h <= n, h++,
        A1 = Expand[A.B];
        a = Expand[-1/h*Tr[A1]];
        If [a =!= 0, k = h; at = a; Btm2 = B;];
        Btm1 = B;
        B = Expand[A1 + a*ID];
        If [B === 0*IdentityMatrix[n], t = h; Break[];];
      ];
      Return[{k, t, Expand[at], Expand[Btm2]}];
    ];
```

Kada je na ulazu konstantna, polinomijalna ili racionalna matrica A formata $n \times n$, funkcija vraća listu sa elementima k^A, t^A, a^A i B^A respektivno. Funkcija radi kako za polinomne tako i za konstantne matrice.

Funkcija `DegreeEstimatorDrazin[A, i, var]` implementira Algoritam 4.3.2 i procenjuje (daje gornju granicu za) stepen polinoma $a_i^{A(s)}$ i matricu stepena matrice $B_{i-1}^{A(s)}$.

```
DegreeEstimatorDrazin[A_, i_, var_] :=
    Module[{h, j, k, l, m, d1, d2, Bd, ad, AA, Ad, Btm1d, Btm2d, atd, td,
        IDd},
      {d1, d2} = Dimensions[A];
      Ad = MatrixDg[A, var];
      Ad = MultiplyDG[Ad, Transpose[Ad]];
      Bd = MatrixDg[IdentityMatrix[d1], var];

      IDd = Bd; td = -1; l = -1; ad = -\[Infinity];
      For [h = 1, h <= i, h++,

        A1d = MultiplyDG[Ad, Bd];
        ad = Max[Table[A1d[[j, j]], {j, d1}]];

        td = h; atd = ad; Btm2d = Bd;

        Btm1d = Bd;
        Bd = A1d;
        For [j = 1, j <= d1, j++,
          Bd[[j, j]] = Max[Bd[[j, j]], ad];
          ];
        ];
      Return[{atd, Btm2d}];
      ];
```

Glavna petlja u ovoj funkciji (u odnosu na promenljivu `h`) implementira korake 5-10 Algoritma 4.3.1. Pomoćne funkcije `MatrixDg[A, var]` i `MultiplyDG[Ad,Bd]` za izračunavanje matrice stepena matrice $\mathrm{dg}A(s)$ i njegove gornje granice, respektivno, takođe su ovde korišćene.

Funkcija `GeneralInvDrazinPoly[A, var]` implementira Algoritam 4.3.1.

```
GeneralInvDrazinPoly[A_, var_] :=
    Module[{AA, dg, tg, deg, n, m, x, tm, i, h, p, Ta, TB, A1, a, B, t, at,
        Btm1, k1, k, degA, Deg, k2, t1, n1, Tk},
      {n, m} = Dimensions[A];
      degA = MatrixPolyDegree[A, s];
      p = n*degA + 1;

      x = Table[i, {i, 1, p}];

      t = -1; k = -1; n1 = n*degA + 1;
      Ta = Table[0, {i, 1, n1}]; TB = Ta; Tk = Ta;
      For [h = 1, h <= n1, h++,
        A1 = ReplaceAll[A, var -> x[[h]]];
        {k1, t1, a, B} = GeneralInvDrazin[A1];
        Tk[[h]] = k1;
        If [t1 > t, t = t1]; If [k1 > k, k = k1]; p = k*degA + 1;
        If [h > t1*degA + 1, Break[]];
        Ta[[h]] = {x[[h]], a}; TB[[h]] = {x[[h]], B};
      ];

      If [k == 0, Return[{0, t, 0, 0*B}]];
      For [h = 1, h <= p, h++,
        If[Tk[[h]] < k,
          A1 = ReplaceAll[A, var -> x[[h]]];
          B = A1.TB[[h, 2]] + Ta[[h, 2]]*IdentityMatrix[n];
          TB[[h, 2]] = MatrixPower[A1, k - 1 - Tk[[h]]].B;
          Ta[[h, 2]] = 0;
        ];
      ];

      {deg, Deg} = DegreeEstimatorDrazin[A, k, var];
      at = DeXSimpleInterpolation[Ta, deg, var];
      Btm1 = AdvMatrixMinInterpolation[TB, Deg, var];

      Return[{k, t, Expand[at], Expand[Btm1]}];
      ];
```

Ulazni parametar ove funkcije je polinomijalna matrica $A(s)$ formata $n \times n$ (promenljiva s je u kodu označena sa `var`) dok je izlaz lista sa elementima $\kappa = k^{A(s)}$, $\tau = t^{A(s)}$, $a_\kappa^{A(s)}$ i $B_{\kappa-1}^{A(s)}$ (`k1`, `t1`, `a` i `B` u kodu).

Najbolji izbor (saglasno vremenu izvršenja) interpolacionih tačaka je $s_i = i$, što je isto kao u odeljku 4.3.3. Ovde je takođe probano i $s_i = -[\frac{n}{2}] + i$, $s_i = \frac{i}{n}$, itd. Vrednosti κ_i, τ_i, a_i i B_i se računaju primenom Algoritma 4.3.1 (funkcija `GeneralInvDrazin[A]`) za konstantnu matricu $A(s_i)$ (označenu sa `A1`). Ove vrednosti se čuvaju u listama `Ta` i `TB`. Posle toga, gornje granice stepena polinoma $a_\kappa^{A(s)}$ kao i matrice stepena matrice $B_{\kappa-1}^{A(s)}$ se procenjuju koristeći funkciju `DegreeEstimatorDrazin[A, n, var]`.

Pomoćne funkcije

- `SimpleInterpolation[Ta, deg, var]`,

- `AdvMatrixMinInterpolation[TB, Deg, var]`,

opisane u odeljku 4.2.4 su takođe korišćene u funkciji `GeneralInvDrazinPoly`. Napomenimo još jedanput da ove funkcije vrše interpolaciju polinoma $a_\kappa^{A(s)}$ i $B_{\kappa-1}^{A(s)}$, respektivno, i da su zasnovane na Newtonovom interpolacionom metodu.

4.3.4 Rezultati testiranja

Testiraćemo implementacije Algoritma 3.3.3 i Algoritma 4.3.1, poboljšane Algoritmom 4.3.2 na test primerima iz [162] i nekim slučajno generisanim test matricama.

U sledećoj tabeli prikazano je vreme izvršenja funkcija `GeneralInvDrazin` (zasnovane na Algoritmu 3.3.3) i `GeneralInvDrazinPoly` (zasnovane na Algoritmu 4.3.1) primenjenih na test primerima iz [162].

Matrica	Alg 3.3.3	Alg 4.3.1
V_2	0.015	0.015
V_3	0.032	0.078
V_4	0.218	1.047
S_5	0.093	0.188
S_{10}	0.875	1.594
S_{15}	3.687	6.688
S_{20}	10.86	22.37

Ove matrice su veoma retke, tako da je Algoritam 4.3.1 (`GeneralInvDrazinPoly`) sporiji od Algoritma 3.3.3 (`GeneralInvDrazin`).

Primer 4.3.1. Razmotrimo test matricu $S_n(t)$ formata $(2n+1)\times(2n+1)$ iz [162]:

$$S_n(t) = \begin{bmatrix} 1+t & t & t & \cdots & t & t & 1+t \\ t & -1+t & t & \cdots & t & t & t \\ t & t & -1+t & \cdots & t & t & t \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ t & t & t & \cdots & -1+t & t & t \\ t & t & t & \cdots & t & -1+t & t \\ 1+t & t & t & \cdots & t & t & 1+t \end{bmatrix}.$$

Važe sledeće relacije

$$k^{S_n(t)} = 2n, \quad t^{S_n(t)} = 2n+1 \quad a^{S_n(t)} = (-1)^n \cdot 2$$

$$B^{S_n(t)} = \begin{bmatrix} 2nt & -t & t & \cdots & t & -t & -1-(2n-1)t \\ -t & 2+2t & -2t & \cdots & -2t & 2t & -t \\ t & -2t & -2+2t & \cdots & 2t & -2t & t \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ t & -2t & 2t & \cdots & -2+2t & -2t & t \\ t & 2t & -2t & \cdots & -2t & 2+2t & -t \\ -1-(2n-1)t & -t & -t & \cdots & t & -t & 2nt \end{bmatrix}$$

Kao što se vidi, samo $2n+3$ elemenata matrice $B^{S_n(t)}$ ima sve nenula koeficijente dok drugi elementi imaju samo jedan. To je razlog zbog koga je naš algoritam sporiji nego Algoritam 3.3.3 kada se primene na test matrice $S_n(t)$.

Srednje vreme izvršenja (srednje potrebno vreme u sekundama za 10 različitih slučajno generisanih test matrica istog tipa) za oba algoritma je prikazano u sledećim tabelama.

deg A	Alg 3.3.3	Alg 4.3.1
5	4.475	1.7188
6	6.0092	2.3158
7	7.6874	2.9034
8	9.5314	3.6876
9	11.85	4.5624
10	14.1814	5.4844
11	16.5938	6.5156
12	19.516	7.6904
13	22.5784	8.9282
14	26.0284	10.3498
15	29.6314	11.9126

$sp_1(A) = sp_2(A) = 1$
$n = 10,\ rankA = 10$

deg A	Alg 3.3.3	Alg 4.3.1
5	2.615	1.625
6	3.604	2.182
7	4.407	2.765
8	5.447	3.479
9	7.020	4.318
10	7.995	5.208
11	9.370	6.182
12	10.510	7.234
13	12.844	8.489
14	14.161	9.791
15	15.937	11.197

$sp_1(A) = sp_2(A) = 0.7$
$n = 10,\ rankA = 10$

Za guste matrice ($sp_1(A) = sp_2(A) = 1$), Algoritam 4.3.1 je mnogo brži u odnosu na Algoritam 3.3.3, na svim test primerima. Za $sp_1(A) = sp_2(A) = 0.7$, Algoritam 3.3.3 je malo brži, ali još uvek sporiji nego Algoritam 4.3.1.

U sledeće dve tabele prikazana su vremena izvršenja algoritama za retke matrice ($sp_1(A) = sp_2(A) = 0.5$) kao i za guste matrice sa malim rangom.

deg A	Alg 3.3.3	Alg 4.3.1
5	1.526	1.505
6	1.890	1.974
7	2.234	2.448
8	3.005	3.192
9	3.640	3.937
10	4.099	4.604
11	4.849	5.520
12	5.276	6.255
13	6.437	8.463
14	7.953	8.958
15	9.448	10.505

$sp_1(A) = sp_2(A) = 0.5$
$n = 10,\ rankA = 10$

deg A	Alg 3.3.3	Alg 4.3.1
5	1.1686	0.5436
6	1.531	0.6998
7	1.9562	0.8938
8	2.4656	1.0626
9	2.9282	1.2314
10	3.428	1.4158
11	4.0438	1.6282
12	4.6596	1.8656
13	5.3094	2.1094
14	6.0496	2.375
15	6.772	2.6468

$sp_1(A) = sp_2(A) = 1$
$n = 10,\ rankA = 3$

Iz prve tabele može da se zaključi da je Algorithm 3.3.3 brži za retke matrice. Slučaj $sp_1(A) = sp_2(A) = 0.5$ je skoro komplementaran slučaju $sp_1(A) = sp_2(A) = 0.7$. Prema tome za matrice sa jednakim spars brojevima ($sp = sp_1(A) = sp_2(A)$), granična vrednost spars broja kada su algoritmi jednako brzi je aproksimativno 0.6. Napomenimo da kada spars brojevi opadaju (uz uslov $sp = sp_1(A) = sp_2(A)$) vreme izvršenja Algoritma 3.3.3 opada naglo, što nije slučaj sa Algoritmom 4.3.1 (čije vreme izvršenja opada sporo). Ovo može da se objasni činjenicom da vreme izvršenja osnovnih operacija sa polinomima u Algoritmu 3.3.3 zavisi od broja nenula koeficijenata, koji opada linearno sa sp (kada je $\deg A$ konstantno). U Algoritmu 4.3.1, vreme izvršenja interpolacije zavisi od stepena polinoma koji će biti interpolirani, i ono opada sporije sa sp.

Druga tabela pokazuje da rang matrice A ima veliki uticaj na vreme izvršenja oba algoritma. Algoritam 4.3.1 je takođe brži nego Algoritam 3.3.3 za matrice sa malim rangom.

Razmotrimo sada dva slučaja kada spars brojevi nisu jednaki.

$\deg A$	Alg 3.3.3	Alg 4.3.1
5	0.404	1.277
10	0.658	2.501
15	0.794	3.375

$sp_1(A) = 0.1$, $sp_2(A) = 1$
$n = 10$, $rankA \leq 10$

$\deg A$	Alg 3.3.3	Alg 4.3.1
5	1.166	1.276
7	1.781	2.229
10	3.150	4.182

$sp_1(A) = 1$, $sp_2(A) = 0.1$
$n = 10$, $rankA \leq 10$

n	Alg 3.3.3	Alg 4.3.1
5	0.398	0.347
6	1.027	0.699
7	2.257	1.312
8	4.582	2.210
9	8.437	3.601
10	14.656	5.593
11	23.750	8.425
12	37.499	12.030

$sp_1(A) = 1$, $sp_2(A) = 1$
$\deg A = 10$, $rankA = n$

Kada je jedan od spars brojeva mali, Algoritam 3.3.3 je takođe brži nego Algoritam 4.3.1. Napomenimo da se vreme izvršenja Algoritma 3.3.3 naglo smanjuje kada je neki od brojeva $sp_1(A)$ ili $sp_2(A)$ mali. U slučaju interpolacije (Algoritam 4.3.1) $sp_2(A)$ skoro da ne utiče na vreme izvršenja, što nije slučaj sa $sp_1(A)$. Kada je broj $sp_1(A)$ mali, stepen matrice $\mathrm{dg}A(s)$ ima veliki broj elemenata jednakih $-\infty$. Takođe, isto važi za izlaznu matricu Algoritma 4.3.2 (funkcija `DegreeEstimatorDrazin[A, i, var]`). Ovo ubrzava proces interpolacije matrice (funkcija `AdvMatrixMinInterpolation[TB, Deg, var]`).

U poslednjoj tabeli prikazano je vreme izvršenja za oba algoritma za $n = 5, \dots, 12$. Koristeći linearnu interpolaciju u log-log skali određujemo aproksimativne složenosti: $\mathcal{O}(n^{5.19})$ i $\mathcal{O}(n^{4.06})$ za Algoritam 3.3.3 i Algoritam 4.3.1, respektivno. Ovo potvrđuje ispravnost teorijski određenih složenosti. Takođe, koristićemo sličan postupak za procenu složenosti kao funkcije od $d = \deg A$. Sve tabele daju približno jednake složenosti $\mathcal{O}(d^{1.7})$ za oba algoritma. Ovim se takođe potvrđuju teorijski određene složenosti. Na kraju ovog odeljka ponovo navodimo krajnje izraze za složenost Algoritma 3.3.3 i Algoritma 4.3.1: $\mathcal{O}(n^5 \cdot d^2)$ i $\mathcal{O}(n^4 \cdot d^2)$ respektivno.

4.4 Interpolacioni metod za računanje različitih generalisanih inverza polinomijalnih matrica

U ovom poglavlju nastavićemo izlaganje iz prethodna dva poglavlja konstruisanjem algoritma za izračunavanje različitih generalisanih inverza date polinomijalne matrice, na osnovu metoda

Leverrier-Faddeeva. Ovaj algoritam je zasnovan na konačnom algoritmu za izračunavanje generalisanih inverza date polinomijalne matrice, koji je uveden u [134]. Na osnovu slične ideje, biće uvedeni metodi za izračunavanje ranga i indeksa polinomijalne matrice. Svi algoritmi su implementirani u simboličkom programskom jeziku MATHEMATICA, i testirani na više različitih klasa test primera.

Ovo poglavlje predstavlja naše originalne rezultate i zasnovano je na našem radu [102]. Kao i u prethodna dva poglavlja, pretpostavićemo da su elementi matrice A polinomi sa **realnim** koeficijentima.

4.4.1 Generalisani metod Leverrier-Faddeevog tipa za polinomijalne matrice

Kao i prethodna dva poglavlja i ovo poglavlje počinje analizom složenosti Algoritma 3.3.4 kada su ulazne matrice R i T polinomijalne matrice, tj. kada je $R(s), T(s) \in \mathbb{R}^{n\times m}[s]$. Da bi uprostili notaciju uvešćemo sledeće oznake.

Definicija 4.4.1. *Neka su $k^{R,T}$, $a_i^{R,T}$ i $B_i^{R,T}$, vrednosti k, a_i i B_i respektivno, za $i = 0,\dots,n$ kada su ulazni parametri Algoritma 3.3.4 matrice R i T (bilo polinomijalne, bilo konstantne). Takođe, označimo $a^{R,T} = a^{R,T}_{k^{R,T}}$ i $B^{R,T} = B^{R,T}_{k^{R,T}-1}$.*

Predhodna definicija je slična sa Definicijom 4.2.1 i Definicijom 4.3.1. Sledeća lema pokazuje osnovna svojstva $a_i^{R,T}$ i $B_i^{R,T}$.

Lema 4.4.1. *Neka su R i T konstantne ili polinomijalne $n\times m$ matrice. Označimo $A' = TR^T$. Tada važi:*

(a) $B^{R,T}_{k^{R,T}+i-1} = (A')^{i-1}\left(A'B^{R,T} + a^{R,T}I_n\right)$ *za sve $i = 1,\dots,n-k^{R,T}$,*

(b) *ako su $R = R(s)$ i $T = T(s)$ polinomijalne matrice, tada takođe važi $\deg B_i^{R(s),T(s)}(s) \le i\cdot d'$ i $\mathrm{dg} a_i^{R(s),T(s)}(s) \le i\cdot d'$ za svako $i = 0,1,\dots,n$.*

Sada ćemo izložiti analizu složenosti Algoritma 3.3.4. Postupak je sličan kao i u slučaju Algoritma 3.3.2 i Algoritma 3.3.3. Zbog toga nećemo ići dublje u detalje.

Neka je $A'(s) = T(s)R(s)^T$ i $d' = \deg A'(s)$. Telo petlje u koracima 4-8 ima složenost $\mathcal{O}(n^3\cdot id'\cdot d')$ u i-tom prolasku kroz petlju, ako pretpostavimo da je složenost množenja dve matrice $\mathcal{O}(n^3)$. Prema tome, ukupna složenost petlje u koracima 4-8 je

$$\mathcal{O}\left(n^3 d'^2 \sum_{i=1}^{n} i\right) = \mathcal{O}(n^5\cdot d'^2). \tag{4.5}$$

Složenost ostalih koraka je znatno niža tako da izraz (4.5) daje ukupnu složenost Algoritma 3.3.4 kada su ulazne matrice R i T polinomijalne. U praksi je vreme izvršenja Algoritma 3.3.2 nešto manje (nemaju svi elementi matrica $B_j(s)$, $A_j(s)$ i $A'(s)$ maksimalan stepen), ali je ipak dosta veliko.

4.4.2 Glavna teorema i interpolacioni algoritam

U sledećoj teoremi odredićemo dovoljan broj interpolacionih tačaka za računanje vrednosti $k^{T(s),R(s)}$, polinomijalne matrice $B^{R(s),T(s)}$ i polinoma $a^{R(s),T(s)}$.

Teorema 4.4.2. *Neka je $T(s), R(s) \in \mathbb{R}[s]^{n\times m}$, $A'(s) = T(s)R(s)^T$, $\kappa = k^{A(s)}$ i $d' = \deg A'(s)$. Važe sledeća tvrdjenja:*

(a) *Neka su s_i, $i = 0, \ldots, n \cdot d'$ međusobno različiti realni brojevi. Tada je*

$$\kappa = \max\{k^{R(s_i),T(s_i)} \mid i = 0, \ldots, n \cdot d'\}.$$

(b) *Polinomijalna matrica $B^{R(s),T(s)}$ i polinom $a^{R(s),T(s)}$ mogu da se rekonstruišu koristeći skup vrednosti $B^{R(s_i),T(s_i)}$ i $a^{R(s_i),T(s_i)}$, $i = 0, \ldots, k^{R(s),T(s)} \cdot d'$.*

Dokaz.

(a) Neka su s_i, $i = 0, \ldots, n \cdot d'$ međusobno različiti realni brojevi i $k' = \max\{k^{R(s_i),T(s_i)} \mid i = 0, \ldots, n \cdot \deg A'(s)\}$. Pokazaćemo da je $k' = \kappa$.

Pretpostavimo da je $a_\kappa^{R(s),T(s)}(s_i) = 0$ za sve $i = 0, \ldots, n \cdot d'$. Prema Algoritmu 3.3.4, stepen polinoma $a_\kappa^{R(s),T(s)}(s)$ ograničen je sa $\kappa \cdot d'$. Pošto je $\kappa \cdot d' \le n \cdot d'$, imamo $a_\kappa^{R(s),T(s)}(s) = 0$, što je u kontradikciji sa definicijom κ. Tada važi

$$(\exists i_0 \le n)(a_\kappa^{R(s_{i_0}),T(s_{i_0})} = a_\kappa^{R(s_{i_0}),T(s_{i_0})}(s_{i_0}) \neq 0),$$

odakle sledi $\kappa \le k^{R(s_{i_0}),T(s_{i_0})} \le k'$.

S druge strane, iz definicije κ imamo $a_{\kappa+t}^{R(s),T(s)}(s) = 0$ za sve $t = 1, \ldots, n - \kappa$. Pošto je jednakost $a_{\kappa+t}^{R(s_i),T(s_i)} = a_{\kappa+t}^{R(s),T(s)}(s_i) = 0$ zadovoljena za sve $i = 0, \ldots, n \cdot d'$, može da se zaključi da je $a_{\kappa+t}^{R(s_i),T(s_i)} = 0$. Prema tome, $k^{R(s_i),T(s_i)} \le \kappa$ važi za sve $i = 0, \ldots, n \cdot d'$, i dobijamo $k' \le \kappa$. Ovim je završen deo dokaza pod **(a)**.

(b) Označimo $B_i' = B^{R(s_i),T(s_i)}$ i $a_i' = a^{R(s_i),T(s_i)}$. Lako može da se pokaže da se vrednosti $B^{R(s),T(s)}(s_i)$ i $a^{R(s),T(s)}(s_i)$ mogu izračunati koristeći sledeće relacije

$$B^{R(s),T(s)}(s_i) = B_{\kappa-1}^{R(s_i),T(s_i)} = \begin{cases} A'(s_i)^{\kappa-\kappa_i-1}\left(A'(s_i)B_i' + a_i' I_n\right), & \kappa > \kappa_i \\ B_i', & \kappa = \kappa_i \end{cases}$$

$$a^{R(s),T(s)}(s_i) = \begin{cases} a_i', & \kappa_i = \kappa, \\ 0, & \kappa_i < \kappa. \end{cases}$$

Sada znamo vrednosti polinoma $B^{R(s),T(s)}$ i $a^{R(s),T(s)}$ u $\kappa \cdot d' + 1$ različitih tačaka. Iz $\deg B^{R(s),T(s)} \le (\kappa - 1) \cdot d'$ i $\mathrm{dg} a^{R(s),T(s))} \le \kappa \cdot d'$ sledi da polinomi $B^{R(s),T(s)}$ i $a^{R(s),T(s)}$ mogu da se rekonstruišu iz skupa tačaka $B^{R(s),T(s)}(s_i)$ i $a^{R(s),T(s)}(s_i)$ $(i = 0, \ldots, \kappa \cdot d')$ koristeći interpolaciju.
$\square$

Teorema 4.4.2 daje glavnu ideju za konstrukciju interpolacionog metoda koji je realizovan Algoritmom 4.4.1.

Algoritam 4.4.1 Interpolacioni algoritam Leverrier-Faddeevog tipa za izračunavanje različitih generalisanih inverza

Input: Matrice $R(s), T(s) \in \mathbb{R}[s]^{n\times m}$ i pozitivni celi brojevi $e \in \mathbb{N}$.

$A'(s) := T(s)R(s)^T$
$d' := \deg A'(s)$
Odredi različite realne brojeve $s_0, s_1, \dots, s_{nd'} \in \mathbb{R}$
for $i := 0$ to $n \cdot d'$ **do**
  Primeni Algoritam 3.3.4 na ulazne matrice $R(s_i)$ i $T(s_i)$, bez izvršavanja koraka **return** (korak 10).
  $\kappa_i := k^{R_i,T_i}$
  $B'_i := B^{R_i,T_i}_{\kappa_i-1}$
  $a'_i := a^{R_i,T_i}_{\kappa_i}$
end for
$\kappa := \max\{\kappa_i \mid i = 0, \dots, n \cdot d'\}$
if $\kappa = 0$ **then**
  return $X_e(s) = \mathbb{O}$
else
  for $i := 0$ to $\kappa \cdot d'$ **do**
    $A'_i := A'(s_i)$
    $B_i := \begin{cases} {A'_i}^{\kappa-\kappa_i-1}\left(A'_i B'_i + a'_i I_n\right), & \kappa > \kappa_i \\ B'_i, & \kappa = \kappa_i \end{cases}$
    $a'_i := \begin{cases} 0, & \kappa > \kappa_i \\ a_i, & \kappa = \kappa_i \end{cases}$
  end for
  Interpoliraj polinom $a^{R(s),T(s)}$ i polinomijalnu matricu $B^{R(s),T(s)}$ koristeći parove (s_i, a_i) i (s_i, B_i), $i = 0, \dots, \kappa \cdot d'$ kao interpolacione tačke.
  return $X_e(s) := (-1)^e a_k(s)^{-e} R(s)^T B_{k-1}(s)^e$
end if

Treba napomenuti da se interpolacija matrice u koraku 24 izvršava interpolirajući svaki element $\left(B_{\kappa-1}^{R(s),T(s)}\right)_{pq}$ sa vrednostima $(B_i)_{pq}$ za $i = 0, \ldots, \kappa \cdot d'$.

U koracima 15-23, ažuriramo samo prvih $\kappa \cdot d'$ matrica B_i i brojeva a_i, jer je to dovoljno za korak 24.

Sada ćemo izvršiti analizu složenosti Algoritma 4.4.1. Prvo, imamo petlju od $d + 1$ ciklusa u koracima 5-10. U svakom ciklusu, vrednosti a_i, B_i i κ_i se računaju koristeći Algoritam 3.3.4 za konstantne matrtice A_i. Podsetimo se da je složenost Algoritma 3.3.4 za konstantne matrice $\mathcal{O}(n^4)$. Prema tome, složenost petlje u koracima 5-10 je $\mathcal{O}(n^4 \cdot d) = \mathcal{O}(n^5 \cdot d')$. U koracima 15-23, matrice B_i se računaju u vremenu $\mathcal{O}(n \cdot n^3 \cdot \log(\kappa - \kappa_i)) = \mathcal{O}(n^4 \cdot \log(n \cdot d'))$. Pretpostavljamo da se stepenovanje matrica obavlja u logaritamskom vremenu $\mathcal{O}(\log(m))$ koristeći rekurzivne formule $A^{2l} = (A^l)^2$ i $A^{2l+1} = (A^l)^2 A$. Na kraju, složenost koraka 24 (interpolacija) je $\mathcal{O}(n^2 \cdot d^2) = \mathcal{O}(n^4 \cdot d'^2)$, kada koristimo Newtonov interpolacioni metod. Prema tome, složenost celog algoritma je $\mathcal{O}(n^4 \cdot d'^2 + n^5 \cdot d')$.

Dobijena složenost je bolja (ali ne mnogo) od složenosti Algoritam 3.3.4 za polinomijalne matrice. Međutim, kako ćemo pokazati u zadnjem odeljku, u praksi je Algoritam 4.4.1 mnogo bolji nego Algoritam 3.3.4, posebno za guste matrice. Takođe, oba algoritma obično ne dostižu svoju maksimalnu složenost.

Algoritam procene stepena za $B_i^{R(s),T(s)}(s)$ i $a_i^{R(s),T(s)}(s)$, $i = 0, 1, \ldots, n \cdot d'$ može da se konstruiše slično kao u prethodna dva poglavlja.

4.4.3 Izračunavanje ranga i indeksa polinomijalnih matrica

Za izračunavanje nekih generalisanih inverza koristeći Teoremu 4.4.2, potrebno je da se izračuna rang i indeks nekih matrica. Ove matrice su polinomijalne, pa se primenom poznatih metoda za izračunavanje ranga i indeksa, kao međurezultati dobijaju racionalne matrice. Zato su vremena izvršenja ovih metoda veoma velika. Naši algoritmi rade samo sa konstantnim matricama i vreme izvršenja je drastično smanjeno.

Za izračunavanje ranga polinomijalnih matrica, koristićemo sličan metod koji je korišćen u odeljku 4.2.2. Sledeća teorema i njena posledica pokazuju primenu Leverrier-Faddeevog metoda i prethodno razmotrenog interpolacionog metoda na ovaj problem.

Teorema 4.4.3. *Neka je* $A \in \mathbb{R}^{n \times n}$. *Tada važi* $\operatorname{rank} A = k^{A,I_n}$.

Dokaz. Neka važi $A' = AI_n^T = A$. Označimo $k = k^{A,I_n}$, $r = \operatorname{rank} A'$ i $a_i = a_i^{A,I_n}$. Razmotrimo karakterističan polinom $P_A(\lambda)$ matrice A. Prema definiciji broja k i zbog činjenice da je $A' = A$ važi

$$P_A(\lambda) = \det(A - \lambda I_n) = a_k \lambda^{n-k} + a_{k-1} \lambda^{n-k+1} + \ldots + a_1 \lambda^{n-1} + \lambda_n. \tag{4.6}$$

Izraz (4.6) ekvivalentan je izrazu

$$P_A(\lambda) = \lambda^{n-k} \sum_{i=0}^{k} a_i \lambda^{k-i}. \tag{4.7}$$

Podsetimo se da je $a_k \neq 0$ i da polinom $P_1(\lambda) = \sum_{i=0}^{k} a_i \lambda^{k-i}$ nema nule $\lambda = 0$.

Predstavićemo matricu A u obliku $A = P^{-1}DP$ gde je D Jordanova normalna forma matrice A a P je regularna matrica. Preuredimo sopstvene vrednosti $\lambda_1, \ldots, \lambda_n$ tako da je prvih r sopstvenih vrednosti različito od nule. Tada važi

$$p_A(\lambda) = p_D(\lambda) = \lambda^{n-r} \prod_{i=1}^{r} (\lambda - \lambda_i)\,, \tag{4.8}$$

Upoređujući (4.7) i (4.8) zaključujemo da je $k = r$, odnosno da je $\mathrm{rank}A = k = k^{R,T}$. □

Posledica 4.4.4. *Neka je $A(s) \in \mathbb{R}[s]^{n\times n}$ i neka su s_i, $i = 0, 1, \ldots, n \cdot \deg A(s)$ različiti realni brojevi. Tada $\mathrm{rank}A(s)$ može da se izračuna na sledeći način:*

$$\mathrm{rank}A(s) = \max\{\mathrm{rank}A(s_i) \,|\, i = 0, \ldots, n \cdot \deg A(s)\} \tag{4.9}$$

Dokaz. Koristeći Teoremu 4.4.3 imamo da u slučaju $R(s) = A(s)$ i $T(s) = I_n$ važi $\kappa = \mathrm{rank}A(s)$ i $\kappa_i = \mathrm{rank}A'_i = \mathrm{rank}A(s_i)$. Prema tome, zaključak je ispravan na osnovu dela **(a)** Teoreme 4.4.2. □

Napomenimo da u formuli (4.9) možemo da koristimo bilo koji metod za izračunavanje ranga konstantnih matrica. Na primer, ako se koristi Gaussov metod eliminacije, složenost je $\mathcal{O}(n \cdot \deg A(s) \cdot n^3) = \mathcal{O}(n^4 \cdot \deg A(s))$. Ako se vrednosti $\mathrm{rank}A(s_i)$ računaju koristeći Leverrier-Faddeev metod (Algoritam 4.4.1) tada je složenost $\mathcal{O}(n^5 \cdot d)$.

Koristeći Teoremu 4.4.3 i Posledicu 4.4.4, slično kao u slučaju MP inverza, blago poboljšanje Algoritma 4.4.1 može da se postigne preračunavanjem vrednosti $\kappa = \max\{\mathrm{rank}A'_i \,|\, i = 0, \ldots, d\}$ $(A'_i = A'(s_i))$. Ova modifikacija je iskorišćena u implementaciji Algoritma 4.4.1.

Algoritam 4.4.2 za izračunavanje ranga date polinomijalne matrice $A(s)$ je zasnovan na Posledici 4.4.4.

Algoritam 4.4.2 Izračunavanje ranga kvadratne polinomijalne matrice

Input: Polinomijalna matrica $A(s) \in \mathbb{R}[s]^{n\times n}$

1: $d := n \cdot \deg A(s)$
2: Odredi različite realne tačke $s_0, \ldots, s_d$.
3: **for** $i := 0$ to d **do**
4: $\quad\kappa_i := \mathrm{rank}A(s_i)$ (izračunati $\mathrm{rank}A(s_i)$ koristeći bilo koji metod za izračunavanje ranga konstantne matrice)
5: **end for**
6: **return** $\mathrm{rank}A(s) := \max\{\mathrm{rank}A(s_i) \,|\, i = 0, \ldots, d\}$

Sada ćemo opisati algoritam za izračunavanje indeksa date polinomijalne matrice $A(s)$. Definišimo vrednosti $t^{R,T}$ kao $t^{R,T} = \min\{r \mid B_r^{R,T} = \mathbb{O}\}$. Evidentno je da važi $t^{A,I_n} = t^A$ (t^A je određeno Definicijom 4.3.1) odakle sledi da je $t^{A,I_n} = n - \mathrm{ind}A$.

Teorema 4.4.5. *Za proizvoljne matrice $R(s), T(s) \in \mathbb{R}[s]^{n\times m}$, i različite realne brojeve $s_0, \dots, s_d$, gde je $d = nd' = n \cdot \deg\big(T(s)R(s)^T\big)$, važi:*

$$t^{R(s),T(s)} = \max\{t^{R(s_i),T(s_i)} \mid i = 0, \dots d\} \tag{4.10}$$

Dokaz. Dokaz je sličan dokazu dela **(b)** Teoreme 4.3.2. □

Sledeća posledica konačno uspostavlja algoritam za izračunavanje $\mathrm{ind}A(s)$, za datu kvadratnu polinomijalnu matricu $A(s) \in \mathbb{R}[s]^{n\times n}$.

Posledica 4.4.6. *neka je $A(s) \in \mathbb{R}[s]^{n\times n}$ data polinomijalna matrica. Tada* $\mathrm{ind}A(s)$ *može da se izračuna koristeći sledeću formulu*

$$\mathrm{ind}A(s) = \min\{\mathrm{ind}A(s_i) \mid i = 0, \dots, n \cdot \deg A(s)\} \tag{4.11}$$

Dokaz. Pošto je $\mathrm{ind}A(s) = n - t^{A,I_n}$ i $\mathrm{ind}A(s_i) = n - t^{A(s_i),I_n}$, zaključak sledi iz Teoreme 4.4.5. □

Kao i u prethodnom slučaju, možemo da koristimo bilo koji metod za izračunavanje indeksa konstantnih matrica $A(s_i)$. Jedan od metoda može da se izvede direktno iz Teoreme 4.4.5. Složenost takvog metoda je $\mathcal{O}(n^4 \cdot nd') = \mathcal{O}(n^5 \cdot \deg A(s))$.

Algoritam 4.4.3 za izračunavanje indeksa date polinomijalne matrice $A(s)$ je zasnovan na Posledici 4.4.6.

Algoritam 4.4.3 Izračunavanje indeksa kvadratne polinomijalne matrice

Input: Polinomijalna matrica $A(s) \in \mathbb{R}[s]^{n\times n}$.

$d := n \cdot \deg A(s)$
Odredi različite realne tačke $s_0, \dots, s_d$.
for $i := 0$ to d **do**
  $\tau_i := \mathrm{ind}A(s_i)$ (izračunaj $\mathrm{ind}A(s_i)$ koristeći neki metod za izračunavanje indeksa konstantnih matrica)
end for
return $\mathrm{ind}A(s) := \min\{\mathrm{ind}A(s_i) \mid i = 0, \dots, d\}$

4.4.4 Implementacija

Svi algoritmi su implementirani u simboličkom programskom jeziku MATHEMATICA. Funkcija `RTGeneralInv[A, p]` implementira blago modifikovanu verziju Algoritma 3.3.4.

```
RTGeneralInv[R_, T_, kk_] :=
    Module[{AA, n, m, t, l, h, a, A1, B, k, at, Btm1, Btm2, ID},
      AA = T.Transpose[R];
      {n, m} = Dimensions[AA];
      ID = IdentityMatrix[n];
      B = IdentityMatrix[n]; t = -1; l = -1; a = 1;
```

```
      Btm1 = B;
      For [h = 1, h <= n, h++,
        A1 = Expand[AA.B];
        a = Expand[-1/h*Tr[A1]];
        If [a =!= 0, t = h; at = a; Btm2 = B;];
        Btm1 = B;
        B = Expand[A1 + a*ID];
        If [h == kk, Return[{t, Expand[a], Expand[Btm1]}];];
      ];
      Return[{t, Expand[at], Expand[Btm2]}];
    ];
```

Za ulazne matrice $R, T \in \mathbb{R}^{n\times m}$ i pozitivan ceo broj p (`kk` u kodu) funkcija vraća listu sa elementima p, $a_p^{R,T}$ i $B_{p-1}^{R,T}$ respektivno, ako je $0 \le p \le n$. U suprotnom, vraća se lista sa elementima $k^{R,T}$, $a^{R,T}$ i $B^{R,T}$. Funkcija radi za konstantne, racionalne i polinomijalne matrice.

Funkcije `PolyMatrixRank[A, var]` i `PolyMatrixIndex[A, var]` implementiraju Algoritam 4.4.2 i Algoritam 4.4.3, respektivno. U funkciji `PolyMatrixRank[A, var]` za izračunavanje ranga konstantnih matrica koristi se funkcija `MatrixRank[A]` ugrađena u programski paket MATHEMATICA. Slično, u funkciji `PolyMatrixIndex[A, var]` koristi se funkcija `MatrixIndex[A]` zasnovana na modifikovanoj verziji Algoritma 3.3.4.

```
PolyMatrixRank[A_, var_] := Module[{r, r1, n, m, x},
      Print[Dimensions[A]];
      {n, m} = Dimensions[A];
      p = 1 + n*MatrixDg[A, var];
      x = Table[i, {i, 1, p}];
      r = 0;
      For [h = 1, h <= p, h++,
        r1 = MatrixRank[ReplaceAll[A, var -> x[[h]]]];
        If [r1 > r, r = r1];
      ];
      Return[r];
    ];
```

Funkcija `RTGeneralInvPoly[A, var]` implementira malu modifikaciju Algoritma 4.4.1 (koristeći preračunavanje ranga matrice definisano Teoremom 4.4.3, tj. Algoritmom 4.4.2).

```
RTGeneralInvPoly[R_, T_, var_] :=
    Module[{R1, T1, dg, tg, deg, n, m, x, tm, i, h, p, Ta, TB, A1, a, B, t, at, Btm1,
        r1, r, degA, Deg},

        AA = Expand[T.Transpose[R]];
        {n, m} = Dimensions[AA];
        degA = MatrixPolyDegree[AA, var];

        p = n*degA + 1;
        x = Table[i, {i, 1, p}];

        r = PolyMatrixRank[AA, var];

        tm = -1; tg = 0;
```

```
        p = r*degA + 1;
        Ta = Table[0, {i, 1, p}]; TB = Table[0, {i, 1, p}];
        For [h = 1, h <= p, h++,
          R1 = ReplaceAll[R, var -> x[[h]]];
          T1 = ReplaceAll[T, var -> x[[h]]];
          {t, a, B} = RTGeneralInv[R1, T1, r];
          Ta[[h]] = {h, a}; TB[[h]] = {h, B};
        ];

       {deg, Deg} = DegreeEstimator[A, r, var];
       at = SimpleInterpolation[Ta, deg, var];
       Btm1 = AdvMatrixMinInterpolation[TB, Deg, var];

       Return[{Expand[at], Expand[Btm1]}];
   ];
```

Ulaz za ovu funkciju su polinomijalne matrice $R(s)$ i $T(s)$ u odnosu na promenljivu s (u kodu, simbolička promenljiva s označena je sa `var`). Dimenzije matrica $R(s)$ i $T(s)$ su jednake n i m respektivno. Funkcija vraća listu elemenata $\kappa = k^{R(s),T(s)}$, $a^{R(s),T(s)}$ i $B^{R(s),T(s)}$. U ovoj implementaciji koristili smo $s_i = i$ kao interpolacione tačke (takođe, kao i u prethodna dva poglavlja ovo je bio najbolji izbor između svih drugih koje smo isprobali).

Kao i u prethodna dva poglavlja, unutar funkcije `RTGeneralInvPoly` korišćene su naše pomoćne funkcije

- `SimpleInterpolation[Ta, deg, var]`,

- `AdvMatrixMinInterpolation[TB, Deg, var]`,

koje vrše interpolaciju polinoma $a_{\kappa}^{R(s),T(s)}$ i $B_{\kappa-1}^{R(s),T(s)}$, respektivno, kroz izračunate tačke. Obe funkcije koriste funkciju `InterpolatingPolynomial[T, var]`, koja je ugrađena u programski paket MATHEMATICA , a zasniva se na Newtonovom interpolacionom metodu.

4.4.5 Rezultati testiranja

Implementacije Algoritma 3.3.4 i Algoritma 4.4.1 (poboljšan algoritmom procene stepena) su testirane sa test primerima iz [162] i sa nekim slučajno generisanim test matricama. Takođe su testirani Algoritam 4.4.2 i Algoritam 4.4.3 na slučajno generisanim test matricama.

U sledećoj tabeli je prikazano vreme izvršenja funkcija `RTGeneralInv` i `RTGeneralInvPoly` za test primere iz [162]. U tim primerima, ulaz funkcije je bio $R(s) = T(s) = A(s)$, $e = 1$ a izlaz, saglasno delu **(1)** Teoreme 3.3.6, Moore-Penroseov inverz $A^{\dagger}(s)$. Sva vremena su u sekundama.

Matrica	Alg 3.3.4	Alg 4.4.1
S_3	0.049	0.070
S_6	0.42	0.67
S_{10}	2.04	5.59
V_4	0.08	1.3
V_5	0.63	16.2
H_3	0.01	0.03
H_6	0.04	0.2
H_{10}	0.5	2.01

Ove matrice su veoma retke, tako da je Algoritam 4.4.1 (`RTGeneralInvPoly`) sporiji nego Algoritam 3.3.4 (`RTGeneralInv`).

Kao što će biti pokazano, spars brojevi sp_1 i sp_2 utiču na vremena izvršenja Algoritma 3.3.4 i Algoritma 4.4.1. Slučajne test matrice su generisane koristeći našu funkciju `RandomMatrix[n, deg, prob1, prob2, var]`. Podsetimo se da ova funkcija generiše slučajnu $n \times n$ polinomijalnu matricu A u odnosu na promenljivu `var`, čiji je stepen ($\deg A$) jednak `deg` a dva spars broja ($sp_1(A)$ i $sp_2(A)$) su jednaka `prob1` i `prob2`. U sledećim tabelama predstavljena su srednja vremena izvršenja (srednja vrednost vremena izvršenja za 10 različitih slučajno generisanih test matrica istog tipa) oba algoritma za $n = 5, 6, 7$ i $\deg A = 3, 4, 5$. U svim slučajevima računat je Drazinov inverz koristeći deo **(2)** Teoreme 3.3.6. Sve matrice imaju konstantan indeks, tj. $\text{ind}A(s) = 1$.

n	$\deg A$	Alg 3.3.4	Alg 4.4.1
5	3	1.8	1.01
5	4	2.05	1.6
5	5	2.75	2.18
6	3	3.5	1.9
6	4	5.2	3.29
6	5	7.6	4.75
7	3	7.92	3.7
7	4	11.9	6.8
7	5	15.7	8.7

$sp_1(A) = sp_2(A) = 1$

n	$\deg A$	Alg 3.3.4	Alg 4.4.1
5	3	0.69	0.61
5	4	1.27	1.03
5	5	2.01	1.5
6	3	0.79	0.70
6	4	2.2	1.8
6	5	2.6	2.0
7	3	1.24	1.09
7	4	5.4	3.2
7	5	6.9	4.8

$sp_1(A) = sp_2(A) = 0.5$

Može se napomenuti da kada je matrica A gusta, Algoritam 4.4.1 je uvek brži nego Algoritam 3.3.4. Ova razlika je veća kada su spars brojevi veći. Vredi napomenuti da matrice međurezultati obično imaju veće spars brojeve (osnovne matrične operacije obično povećavaju spars brojeve). Prema tome, situacija u slučaju $sp_1(A) = sp_2(A) = 0.5$ je slična slučaju $sp_1(A) = sp_2(A) = 1$.

Za manje spars brojeve Algoritam 3.3.4 je brži nego Algoritam 4.4.1. Napomenimo da kada se spars brojevi smanjuju (uz uslov $sp = sp_1(A) = sp_2(A)$) vreme izvršavanja Algoritma 3.3.4 naglo opada, što nije slučaj kod Algoritma 4.4.1, gde vreme izvršavanja opada sporije.

Razmotrimo sada slučaj kada spars brojevi $sp_1(A)$ i $sp_2(A)$ nisu jednaki.

n	$\deg A$	Alg 3.3.4	Alg 4.4.1
9	3	1.9	1.0
9	4	3.1	1.9
9	5	4.6	2.6
10	3	3.6	1.6
10	4	5.5	2.9
10	5	7.7	4.0
11	3	5.9	2.7
11	4	8.5	4.1
11	5	14.2	7.05

$sp_1(A) = 1$, $sp_2(A) = 0.1$

n	$\deg A$	Alg 3.3.4	Alg 4.4.1
9	3	0.1	0.25
9	4	0.11	0.21
9	5	0.2	0.44
10	3	0.19	0.48
10	4	0.3	0.59
10	5	0.42	0.81
11	3	0.54	0.72
11	4	0.59	0.78
11	5	0.71	1.77

$sp_1(A) = 0.1$, $sp_2(A) = 1$

Primetimo da se vreme izvršavanja Algoritma 3.3.4 naglo smanjuje kada je bilo $sp_1(A)$ bilo $sp_2(A)$ malo. U slučaju interpolacionog metoda (Algoritam 4.4.1) $sp_2(A)$ skoro da i ne utiče na vreme izvršavanja, što nije slučaj sa $sp_1(A)$. Kada je $sp_1(A)$ malo, matrica stepena $\mathrm{dg}A(s)$ ima veliki broj elemenata jednak $-\infty$.

U sledećoj tabeli prikazano je poređenje rezultata testiranja Algoritma 4.4.2 i Algoritma 4.4.3 sa rezultatima koji su dobijeni direktnom primenom funkcija `MatrixRank` i `MatrixIndex` na polinomijalne matrice. U ovom slučaju, vreme izvršavanja za sve funkcije direktno zavisi od vrednosti ranga i indeksa matrica. Zbog toga su u tabeli dati odnosi odgovarajućih vremena izvršavanja.

n	$\deg A$	$\frac{\text{Alg.4.4.2}}{\texttt{MatrixRank}}$	$\frac{\text{Alg.4.4.3}}{\texttt{MatrixIndex}}$
9	3	1.3	1.0
9	4	2.1	1.6
9	5	4.6	2.3
10	3	3.6	1.1
10	4	5.8	2.3
10	5	8.7	4.2
11	3	9.3	2.3
11	4	10.5	4.6
11	5	16.7	8.95

4.5 Metod pregrađivanja za MP inverze polinomijalnih matrica sa dve promenljive

U ovom poglavlju predložićemo modifikaciju Grevilleovog metoda pregrađivanja (Algoritam 3.4.1) za izračunavanje MP inverza racionalnih i polinomijalnih kompleksnih matrica sa dve promenljive. Ovi rezultati su uopštenje rezultata Stanimirovića i Tasića uvedenih u [135]. Na kraju ovog poglavlja biće dato nekoliko primera.

U ovom poglavlju su izloženi naši originalni rezultati i njegovu osnovu čine naši radovi [100, 101].

4.5.1 Racionalne matrice

Neka je $A(s_1, s_2) \in \mathbb{C}^{m\times n}[s_1, s_2]$ racionalna matrica sa dve promenljive. Pošto Teorema 3.4.1 važi i za racionalne matrice, Grevilleov metod pregrađivanja, uveden u poglavlju 3.4 (Algoritam 3.4.1), može direktno da se primeni na racionalnu matricu $A(s_1, s_2)$. Medjurezultati koji se izračunavaju u Algoritmu 3.4.1, kao što će biti pokazano, u opštem slučaju su racionalne funkcije 4 promenljive $s_1, s_2, \overline{s_1}$ i $\overline{s_2}$. Zbog toga ćemo uvesti nove promenljive $s_3 = \overline{s_2}$ i $s_4 = \overline{s_1}$.

Oznake definisane u poglavlju 3.4 koristimo i u ovom poglavlju. Uvedimo oznaku $S = (s_1, s_2, s_3, s_4)$. Nadalje će sve racionalne i polinomijalne matrice će biti razmatrane kao funkcije od S. Podsetimo se da je $A_i(S)$ submatrica matrice $A(S)$ koja sadrži prvih i kolona ove matrice i da je $a_i(S)$ i-ta kolona matrice $A(S)$,

$$A_i(S) = [A_{i-1}(S) \mid a_i(S)]\,, i = 2, \dots, n, \quad A_1(S) = a_1(S). \tag{4.12}$$

U implementaciji metoda pregadjivanja, koristćemo funkciju `Together` programskog paketa MATHEMATICA da bi omogućili uprošćavanje racionalnih izraza (ova funkcija grupiše racionalne sabirke i skraćuje zajedničke činioce u imeniocu i brojiocu).

4.5.2 Polinomijalne matrice

U ovom odeljku dokazujemo osnovnu teoremu koja se zasniva na Teoremi 3.4.1 i Grevilleovom metodu pregrađivanja (Algorithm 3.4.1).

Svaku matricu $A(s_1, s_2) \in \mathbb{C}^{m\times n}[s_1, s_2]$ posmatramo u polinomijalnom obliku:

$$A(s_1, s_2) = A(S) = \sum_{j_1=0}^{q_1}\sum_{j_2=0}^{q_2}\sum_{j_3=0}^{q_3}\sum_{j_4=0}^{q_4} A_{j_1,j_2,j_3,j_4} s_1^{j_1} s_2^{j_2} s_3^{j_3} s_4^{j_4}, \tag{4.13}$$

gde su A_{j_1,j_2,j_3,j_4} konstantne $m \times n$ matrice.

Da bi pojednostavili dalje izraze uvodimo sledeće oznake. Neka je $S^J = s_1^{j_1} s_2^{j_2} s_3^{j_3} s_4^{j_4}$, $Q = (q_1, q_2, q_3, q_4) = \deg A$ i $J = (j_1, j_2, j_3, j_4)$. Tada sa $\sum\limits_{J=0}^{Q} A_J S^J$ jednostavno označavamo zbir (4.13). Takođe, ako sa $\overline{Q}$ označimo $\overline{Q} = (q_4, q_3, q_2, q_1)$, može lako da se proveri da važi $A^* = \sum\limits_{J=0}^{\overline{Q}} A_J^* S^J$ za svako $A \in \mathbb{C}^{m\times n}[S]$.

Na sličan način, i-tu kolonu matrice $A(s_1, s_2)$ označavamo sa

$$a_i(S) = \sum_{j_1=0}^{q_1}\sum_{j_2=0}^{q_2}\sum_{j_3=0}^{q_3}\sum_{j_4=0}^{q_4} a_{i,j_1,j_2,j_3,j_4} s_1^{j_1} s_2^{j_2} s_3^{j_3} s_4^{j_4} = \sum_{J=0}^{Q} a_{i,J} S^J,\ 1 \le i \le n,$$

gde su a_{i,j_1,j_2,j_3,j_4} konstantni $m \times 1$ vektori. Takođe, prvih i kolona matrice $A(S)$ označavamo sa

$$A_i(S) = \sum_{j_1=0}^{q_1}\sum_{j_2=0}^{q_2}\sum_{j_3=0}^{q_3}\sum_{j_4=0}^{q_4} A_{i,j_1,j_2,j_3,j_4} s_1^{j_1} s_2^{j_2} s_3^{j_3} s_4^{j_4} = \sum_{J=0}^{Q} A_{i,J} S^J,\ 1 \le i \le n,$$

gde su A_{i,j_1,j_2,j_3,j_4} konstantne $m \times i$ matrice.

Očigledno, Algoritam 3.4.1 je primenljiv na polinomijalne matrice $A(s_1, s_2)$. Ta primena Algoritma 3.4.1 dovodi do glavnog rezultata ovog poglavlja.

Teorema 4.5.1. *Neka je $A(S) \in \mathbb{C}^{m\times n}[S]$. MP inverz $A_i^\dagger \in \mathbb{C}^{i\times m}[S]$ prvih i kolona matrice A je oblika*

$$A_i^\dagger(S) = A_i^\dagger(S) = \frac{X_i(S)}{y_i(S)} = \frac{\sum_{J=0}^{Q_i} X_{i,J} S^J}{\sum_{J=0}^{P_i} y_{i,J} S^J} \tag{4.14}$$

gde je $P_i = Q_i + Q$ i $X_i \in \mathbb{C}^{i\times m}[S]$, $y_i \in \mathbb{C}[S]$ može da se izračuna iz X_{i-1}, y_{i-1}, A_{i-1} i A_i koristeći egzaktne rekurentne relacije.

Dokaz.

Dokaz ćemo izvesti matematičkom indukcijom. Bazu indukcije (slučaj $n = 1$) direktno dokazujemo iz koraka 1 Algoritma 3.4.2. Ako je uslov $a_1(S) \neq 0$ ispunjen, imamo

$$A_1^\dagger = \frac{a_1^*(S)}{a_1^*(S)a_1(S)} = \frac{\sum_{K=0}^{\overline{Q}} a_{1,J}^* S^J}{\sum_{K=0}^{\overline{Q}+Q} \sum_{K=0}^{J} a_{1,J-\overline{K}}^* \, a_{1,K} S^J} = \frac{\sum_{K=0}^{Q_1} X_{1,J} S^J}{\sum_{K=0}^{P_1} y_{1,J} S^J}$$

Ako sada stavimo $X_1(S) = \sum_{K=0}^{Q_1} X_{1,J} S^J$ i $y_1(S) = \sum_{K=0}^{P_1} y_{1,J} S^J$, dobijamo $A_1^\dagger$ u obliku (4.14). Ako je $a_1(S) = 0$, očigledno je da $A_1^\dagger$ ima oblik (4.14).

U dokazu induktivnog koraka, koristićemo korake 3-10 Algoritma 3.4.1. Iz koraka 3, direktnim izračunavanjem dobijamo

$$d_i(S) = \frac{X_{i-1}(S)}{y_{i-1}(S)} a_i(S) = \frac{\sum_{J=0}^{Q_{i-1}+Q} \sum_{K=0}^{J} X_{i-1,J-K} \, a_{i,K} S^J}{\sum_{J=0}^{P_{i-1}} y_{i-1,J} S^J} = \frac{\sum_{J=0}^{Q_{i-1}+Q} d_{i,J} S^J}{\sum_{J=0}^{P_{i-1}} y_{i-1,J} S^J}. \tag{4.15}$$

Koristeći smenu $d_i(S) = \sum_{J=0}^{Q_{i-1}+Q} d_{i,J} S^J$, iz koraka 4 računamo koeficijente polinoma $c_i(S) = \sum_{J=0}^{Q_{i-1}+2Q} c_{i-1,J} S^J$ na sledeći način:

$$\begin{aligned} c_i(S) &= \frac{y_{i-1}(S)a_i(S) - A_{i-1}(S)d_i(S)}{y_{i-1}(S)} \\ &= \frac{\sum_{J=0}^{Q_{i-1}+2Q} \left(\sum_{K=0}^{J} y_{i-1,J-K} a_{i,K} - A_{i-1,J-K} d_{i,K}\right) S^J}{\sum_{J=0}^{P_{i-1}} y_{i-1,J} S^J} = \frac{\sum_{J=0}^{Q_{i-1}+2Q} c_{i,J} S^J}{\sum_{J=0}^{P_{i-1}} y_{i-1,J} S^J} \end{aligned} \tag{4.16}$$

Ako je $c_i(S) \neq 0$ tada saglasno koraku 8, obzirom na $P_{i-1} = Q_{i-1} + Q$, imamo

$$
\begin{aligned}
b_i(S) &= \frac{\left(\sum\limits_{J=0}^{Q_{i-1}+2Q} c_{i,J} S^J\right)\left(\sum\limits_{J=0}^{P_{i-1}} y_{i-1,J} S^J\right)^*}{\left(\sum\limits_{J=0}^{Q_{i-1}+2Q} c_{i,J} S^J\right)^*\left(\sum\limits_{J=0}^{Q_{i-1}+2Q} c_{i,J} S^J\right)} \\
&= \frac{\sum\limits_{J=0}^{Q_{i-1}+\overline{Q_{i-1}}+2Q+\overline{Q}} \left(\sum\limits_{K=0}^{J} c_{i,J-K} y^*_{i-1,\overline{K}}\right) S^J}{\sum\limits_{J=0}^{Q_{i-1}+\overline{Q_{i-1}}+2Q+2\overline{Q}} \left(\sum\limits_{K=0}^{J} c^*_{i,J-K} c_{i-1,\overline{K}}\right) S^J} = \frac{\sum\limits_{J=0}^{dgw_i} w_{i,J} S^J}{\sum\limits_{J=0}^{dgv_i} v_{i-1,J} S^J} = \frac{w_i(S)}{v_{i-1}(S)}. \qquad (4.17)
\end{aligned}
$$

S druge strane, u slučaju $c_i(S) = 0$ imamo

$$
\begin{aligned}
b_i(S) &= \frac{\frac{X^*_{i-1}(S) d_i(S)}{y^*_{i-1}(S) y_{i-1}(S)}}{1 + \frac{d^*_i(S) d_i(S)}{y^*_{i-1}(S) y_{i-1}(S)}} \\
&= \frac{\sum\limits_{J=0}^{Q_{i-1}+\overline{Q_{i-1}}+Q+\overline{Q}} \left(\sum\limits_{K=0}^{\overline{J}} X^*_{i-1,J-K} d_{i,\overline{K}}\right) S^J}{\sum\limits_{J=0}^{Q_{i-1}+\overline{Q_{i-1}}+Q+\overline{Q}} \left(\sum\limits_{K=0}^{\overline{J}} (y^*_{i-1,J-K} y_{i,\overline{K}} + d^*_{i,J-K} d_{i,\overline{K}})\right) S^J} \qquad (4.18) \\
&= \frac{\sum\limits_{J=0}^{dgw_i} w_{i,J} S^J}{\sum\limits_{J=0}^{dgv_i} v_{i-1,J} S^J} = \frac{w_i(S)}{v_{i-1}(S)}.
\end{aligned}
$$

U oba slučaja koristili smo pomoćne polinome $w_i(S) \in \mathbb{C}[S]^{m\times 1}$ i $v_i(S) \in \mathbb{C}[S]$ za čije stepene važi $\deg w_i = \deg v_i + \overline{Q}$. Iz koraka 10, primenjujući prethodni rezultat, izračunavamo $A_i^\dagger(S)$:

$$
\begin{aligned}
A_i^{\dagger}(S) &= \begin{bmatrix} A_{i-1}^{\dagger}(S) - d_i(S)b_i(S)^* \\ b_i^*(S) \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{\sum\limits_{J=0}^{Q_{i-1}} X_{i-1,J}S^J}{\sum\limits_{J=0}^{Q_{i-1}+Q} y_{i-1,J}S^J} - \dfrac{\sum\limits_{J=0}^{\overline{dgw_i}+Q+Q_{i-1}} (\sum\limits_{K=0}^{J} D_{i,J-\overline{K}} w_{i-1,J-K}^* S^J)}{\sum\limits_{J=0}^{\overline{dgv_i}+Q+Q_{i-1}} (\sum\limits_{K=0}^{J} v_{i,K}^* y_{i-1,J-\overline{K}} S^J)} \\ \dfrac{\sum\limits_{J=0}^{\overline{dgw_i}} w_{i,J}^* S^J}{\sum\limits_{J=0}^{\overline{dgv_i}} v_{i,J}^* S^J} \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{\sum\limits_{J=0}^{dg\Theta_i} (\sum\limits_{K=0}^{J} (v_{i,K}^* X_{i-1,J-\overline{K}} - d_{i,J-\overline{K}} w_{i,K}^*))}{\sum\limits_{J=0}^{dg\phi_i} (\sum\limits_{K=0}^{J} v_{i,K}^* y_{i-1,J-\overline{K}}) S^J} \\ \dfrac{\sum\limits_{J=0}^{\overline{dgw_i}} w_{i,J}^* S^J}{\sum\limits_{J=0}^{\overline{dgv_i}} v_{i,J}^* S^J} \end{bmatrix} = \begin{bmatrix} \dfrac{\sum\limits_{J=0}^{dg\Theta_i} \Theta_{i,J} S^J}{\sum\limits_{J=0}^{dg\phi_i} \phi_{i,J} S^J} \\ \dfrac{\sum\limits_{J=0}^{\overline{dgw_i}} w_{i,J}^* S^J}{\sum\limits_{J=0}^{\overline{dgv_i}} v_{i,J}^* S^J} \end{bmatrix}.
\end{aligned}
$$

Ovde smo, takođe, koristili pomoćne polinome $\Theta(S) \in \mathbb{C}(S)^{i\times m}$ i $\phi(S) \in \mathbb{C}(S)$ sa koeficijentima određenim u koraku 8. Konačno je:

$$
A_i^{\dagger}(S) = \begin{bmatrix} \dfrac{\sum\limits_{J=0}^{dg\Theta_i+\overline{dgv_i}} (\sum\limits_{K=0}^{J} v_{i,K}^* \Theta_{i,J-\overline{K}}) S^J}{\sum\limits_{J=0}^{dg\phi_i+\overline{dgv_i}} (\sum\limits_{K=0}^{J} v_{i,K}^* \phi_{i,J-\overline{K}}) S^J} \\ \dfrac{\sum\limits_{J=0}^{\overline{dgw_i}+dg\phi_i} (\sum\limits_{K=0}^{J} \phi_{i,J-\overline{K}} w_{i,K}^*) S^J}{\sum\limits_{J=0}^{\overline{dgv_i}+dg\phi_i} (\sum\limits_{K=0}^{J} v_{i,K}^* \phi_{i,J-\overline{K}}) S^J} \end{bmatrix} = \frac{\sum\limits_{J=0}^{Q_i} X_{i,J} S^J}{\sum\limits_{J=0}^{Q_i+Q} y_{i,J} S^J},
$$

gde je

$$
X_{i,J} = \begin{bmatrix} \sum\limits_{K=0}^{J} v_{i,K}^* \Theta_{i,J-\overline{K}} \\ \sum\limits_{K=0}^{J} \phi_{i,J-\overline{K}} W_{i,K}^* \end{bmatrix}, \quad \deg X_i = Q_i = Q_{i-1} + Q + \overline{\deg W_i} + \overline{\deg v_i} \tag{4.19}
$$

$$
y_{i,J} = \sum_{K=0}^{J} v_{i,K}^* \phi_{i,J-\overline{K}}, \quad \deg y_i = P_i = \overline{\deg v_i} + \deg \phi_i = Q_i + Q \tag{4.20}
$$

Ovim je dokaz završen. $\square$

Posledica 4.5.2. *Pod pretpostavkama Teoreme 4.5.1, u slučaju $i = n$ imamo*

$$
A^{\dagger}(s_1, s_2) = A_n^{\dagger}(s_1, s_2) = \frac{\sum\limits_{J=0}^{Q_n} X_{n,J} S^J}{\sum\limits_{J=0}^{P_n} y_{n,J} S^J}
$$

Algoritam 4.5.1 Metod pregrađivanja za polinomijalne matrice sa dve promenljive

Input: Polinomijalna matrica sa dve promenljive $A(S) \in \mathbb{C}^{m\times n}[S]$.

$X_{1,J} := a^*_{1,J}$, za svako $0 \le J \le Q_1 = Q$

$y_{1,J} := \sum\limits_{K=0}^{J} a^*_{i,J-\overline{K}} a_{i,K}$, za svako $0 \le J \le P_1 = Q + \overline{Q}$

for $i := 2$ to n **do**

  $d_{i,J} := \sum\limits_{K=0}^{J} X_{i-1,J-K} A_{i,K}$, za svako $0 \le J \le \deg d_i = Q_{i-1} + Q$

  $c_{i,J} := \sum\limits_{K=0}^{J} (y_{i-1,J-K} a_{i,K} - A_{i-1,J-K} d_{i,K})$, za svako $0 \le J \le \deg c_i = Q_{i-1} + 2Q$.

  if postoji $c_{i,J} \ne 0$ **then**

    $w_{i,J} := \sum\limits_{K=0}^{J} c_{i,J-K} y^*_{i-1,\overline{K}}$, za svako $0 \le J \le \deg w_i = Q_{i-1} + \overline{Q_{i-1}} + 2Q + \overline{Q}$

    $v_{i,J} := \sum\limits_{K=0}^{J} c^*_{i,J-K} c_{i-1,\overline{K}}$, za svako $0 \le J \le \deg v_i = \deg w_i + \overline{Q}$

  else

    $w_{i,J} := \sum\limits_{K=0}^{\overline{J}} X^*_{i-1,J-K} d_{i,\overline{K}}$, za svako $0 \le J \le \deg w_i = Q_{i-1} + \overline{Q_{i-1}} + Q + \overline{Q}$

    $v_{i,J} = \sum\limits_{K=0}^{\overline{J}} (y^*_{i-1,J-K} y_{i,\overline{K}} + d^*_{i,J-K} d_{i,\overline{K}})$, za svako $0 \le J \le \deg v_i = Q_{i-1} + \overline{Q_{i-1}} + Q + \overline{Q}$

  end if

  $\Theta_{i,J} := \sum\limits_{K=0}^{J} v^*_{i,K} X_{i-1,J-\overline{K}} - d_{i,J-\overline{K}} w^*_{i,K}$, za svako $0 \le J \le \deg \Theta_i = \overline{\deg w_i} + Q + Q_{i-1}$

  $\phi_{i,J} := \sum\limits_{K=0}^{J} v^*_{i,K} y_{i-1,J-\overline{K}}$, za svako $0 \le J \le \deg \phi_i = \overline{\deg v_i} + Q + Q_{i-1}$

  $X_{i,J} := \begin{bmatrix} \sum\limits_{K=0}^{J} v^*_{i,K} \Theta_{i,J-\overline{K}} \\ \sum\limits_{K=0}^{J} \phi_{i,J-\overline{K}} w^*_{i,K} \end{bmatrix}$, za svako $0 \le J \le Q_i = Q_{i-1} + Q + \overline{\deg w_i} + \overline{\deg v_i}$

  $y_{i,J} := \sum\limits_{K=0}^{J} v^*_{i,K} \phi_{i,J-\overline{K}}$, za svako $0 \le J \le P_i = Q_i + \overline{Q}$

end for

return $A(s_1,s_2)^\dagger := A_n(s_1,s_2)^\dagger := \dfrac{\sum\limits_{J=0}^{Q_n} X_{n,J} S^J}{\sum\limits_{J=0}^{P_n} y_{n,J} S^J}$

Rekurentne relacije iz Teoreme 4.5.1 su korišćene da bi se došlo do uopštenja Grevilleovog metoda pregrađivanja opisanog u Algoritmu 4.5.1.

U praksi, obično radimo sa polinomijalnim matricama $A(s_1, s_2)$ koje imaju samo nekoliko nenultih koeficijenata. U tom slučaju, Algoritam 4.5.1 nije efikasan zato što mnoge operacije nisu potrebne. Ovaj problem može da se prevaziđe konstruisanjem efikasnih struktura za predstavljanje polinomijalnih matrica.

Ove strukture su definisane u odeljku 4.6.3 i korišćene su za izračunavanje težinskih MP inverza. Takođe, one mogu biti primenjene na Algoritam 4.5.1.

4.5.3 Numerički primeri

Algoritam 4.5.1 smo implementirali u simboličkom programskom paketu MATHEMATICA. Implementacija je testiranana nekoliko test matrica iz [162].

Primer 4.5.1. Posmatrajmo poznatu test matricu S_9 reda 9 iz [162]

$$S_9 = \begin{bmatrix} 1+t & t & t & t & t & t & t & t & 1+t \\ t & -1+t & t & t & t & t & t & t & t \\ t & t & 1+t & t & t & t & t & t & t \\ t & t & t & -1+t & t & t & t & t & t \\ t & t & t & t & 1+t & t & t & t & t \\ t & t & t & t & t & -1+t & t & t & t \\ t & t & t & t & t & t & 1+t & t & t \\ t & t & t & t & t & t & t & -1+t & t \\ 1+t & t & t & t & t & t & t & t & 1+t \end{bmatrix}$$

Primenom našeg programa dobijamo sledeće MP inverze

$$S_9^\dagger = \begin{bmatrix} \frac{1-t}{4} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & \frac{1-t}{4} \\ \frac{t}{2} & -1-t & t & -t & t & -t & t & -t & \frac{t}{2} \\ -\frac{t}{2} & t & 1-t & t & -t & t & -t & t & -\frac{t}{2} \\ \frac{t}{2} & -t & t & -1-t & t & -t & t & -t & \frac{t}{2} \\ -\frac{t}{2} & t & -t & t & 1-t & t & -t & t & -\frac{t}{2} \\ \frac{t}{2} & -t & t & -t & t & -1-t & t & -t & \frac{t}{2} \\ -\frac{t}{2} & t & -t & t & -t & t & 1-t & t & -\frac{t}{2} \\ \frac{t}{2} & -t & t & -t & t & -t & t & -1-t & \frac{t}{2} \\ \frac{1-t}{4} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & \frac{1-t}{4} \end{bmatrix}$$

Primer 4.5.2. Za blok matricu sa dve promenljive V_8 iz [162]

$$V_8 = \begin{bmatrix} x & t & x & t & x & t & x & t \\ t & -x & t & -x & t & -x & t & -x \\ x & t & -x & -t & x & t & -x & -t \\ t & -x & -t & x & t & -x & -t & x \\ x & t & x & t & -x & -t & -x & -t \\ t & -x & t & -x & -t & x & -t & x \\ x & t & -x & -t & -x & -t & x & t \\ t & -x & -t & x & -t & x & t & -x \end{bmatrix}$$

dobijamo

$$V_8^{\dagger} = \begin{bmatrix}
\frac{x}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{x}{t^2+x^2} \\
\frac{t}{t^2+x^2} & \frac{-x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} \\
\frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} \\
\frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} \\
\frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} \\
\frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} \\
\frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} \\
\frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} & -\frac{t}{t^2+x^2} & \frac{x}{t^2+x^2} & \frac{t}{t^2+x^2} & -\frac{x}{t^2+x^2}
\end{bmatrix}$$

Primer 4.5.3. Za jednoparametarsku matricu F_{11} iz [162]

$$F_{11} = \begin{bmatrix}
11+x & 10+x & 9+x & 8+x & 7+x & 6+x & 5+x & 4+x & 3+x & 2+x & 1+x \\
10+x & 10+x & 9+x & 8+x & 7+x & 6+x & 5+x & 4+x & 3+x & 2+x & 1+x \\
9+x & 9+x & 9+x & 8+x & 7+x & 6+x & 5+x & 4+x & 3+x & 2+x & 1+x \\
8+x & 8+x & 8+x & 8+x & 7+x & 6+x & 5+x & 4+x & 3+x & 2+x & 1+x \\
7+x & 7+x & 7+x & 7+x & 7+x & 6+x & 5+x & 4+x & 3+x & 2+x & 1+x \\
6+x & 6+x & 6+x & 6+x & 6+x & 6+x & 5+x & 4+x & 3+x & 2+x & 1+x \\
5+x & 5+x & 5+x & 5+x & 5+x & 5+x & 5+x & 4+x & 3+x & 2+x & 1+x \\
4+x & 4+x & 4+x & 4+x & 4+x & 4+x & 4+x & 4+x & 3+x & 2+x & 1+x \\
3+x & 3+x & 3+x & 3+x & 3+x & 3+x & 3+x & 3+x & 3+x & 2+x & 1+x \\
2+x & 2+x & 2+x & 2+x & 2+x & 2+x & 2+x & 2+x & 2+x & 1+x & x \\
1+x & 1+x & 1+x & 1+x & 1+x & 1+x & 1+x & 1+x & 1+x & x & -1+x
\end{bmatrix}$$

dobijamo njen MP inverz

$$F_{11}^{\dagger} = \begin{bmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -\frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{5}{6} & \frac{7}{9} - \frac{x}{4} & \frac{4}{9} & \frac{1}{9} + \frac{x}{4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{4}{9} & \frac{1}{9} & -\frac{2}{9} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{9} + \frac{x}{4} & -\frac{2}{9} & -\frac{5}{9} - \frac{x}{4}
\end{bmatrix}$$

Primer 4.5.4. Posmatrajmo sada dvoparametarsku matricu B reda 4×3 iz [162]

$$B = 2 \begin{bmatrix}
x+2t & -2x-t & -2x+2t \\
-x-2t & 2x+t & 2x-2t \\
-x+2t & 2x-t & 2x+2t \\
-x+2t & 2x-t & 2x+2t
\end{bmatrix}$$

MP inverz je jednak

$$B^{\dagger} = \begin{bmatrix}
\frac{t+2x}{72tx} & -\frac{t+2x}{72tx} & \frac{1}{72}\left(\frac{2}{t} - \frac{1}{x}\right) & \frac{1}{72}\left(\frac{2}{t} - \frac{1}{x}\right) \\
-\frac{2t+x}{72tx} & \frac{2t+x}{72tx} & \frac{1}{72}\left(-\frac{1}{t} + \frac{2}{x}\right) & \frac{1}{72}\left(-\frac{1}{t} + \frac{2}{x}\right) \\
\frac{-t+x}{36tx} & \frac{t-x}{36tx} & \frac{t+x}{36tx} & \frac{t+x}{36tx}
\end{bmatrix}$$

Primer 4.5.5. Na kraju, formirajmo dvoparametarsku kompleksnu matricu C formatareda 4×3, sličnu test matrici B (i označava $\sqrt{-1}$)

$$C = 2\begin{bmatrix} x+2it & -2x-t & -2ix+2t \\ -x-2t & 2x+t & 2x-2t \\ -x+2it & 2x-t & 2ix+2t \\ -x+2t & 2x-t & 2x+2t \end{bmatrix}$$

MP inverz je jednak

$$\left[\begin{matrix} \frac{(14+14i)t^2+(9-3i)tx+(2+8i)x^2}{-224t^3-104tx^2} & \frac{(14+14i)t^2+(7+i)tx+6x^2}{-224t^3-104tx^2} \\ \frac{(8-4i)t^3+(10+16i)t^2x+(11+5i)tx^2+(1+4i)x^3}{-224t^3x-104tx^3} & \frac{12t^3+(6-8i)t^2x+(3-7i)tx^2-3x^3}{224t^3x+104tx^3} \\ \frac{(-8+i)t^2+(18+12i)tx+(13+13i)x^2}{448t^2x+208x^3} & \frac{(12t^2-(22-20i)tx+(13+13i)x^2}{448t^2x+208x^3} \end{matrix}\right.$$

$$\left.\begin{matrix} \frac{(14+14i)t^2-(9-3i)tx+(2+8i)x^2}{-224t^3-104tx^2} & \frac{(14+14i)t^2-(7+i)tx+6x^2}{224t^3+104tx^2} \\ \frac{(8-4i)t^3-(10+16i)t^2x+(11+5i)tx^2-(1+4i)x^3}{224t^3x+104tx^3} & \frac{12t^3-(6-8i)t^2x+(3-7i)tx^2+3x^3}{224t^3x+104tx^3} \\ \frac{(8-4i)t^2+(18+12i)tx-(13+13i)x^2}{448t^2x+208x^3} & \frac{12t^2+(22-20i)tx+(13+13i)x^2}{448t^2x+208x^3} \end{matrix}\right]$$

Kao što možemo da vidimo iz ovih primera, MP inverzi su racionalne funkcije samo od s_1 i s_2. Zaista, razlomak $\frac{X_i(S)}{y_i(S)}$ može da se redukuje tako da s_3 i s_4 iščeznu. Takođe, napomenimo da isto važi za $\frac{W_i(S)}{v_i(S)}$ i $\frac{\Theta_i(S)}{\phi_i(S)}$. Prema tome možemo da damo sledeću hipotezu.

Hipoteza 4.5.3. *Izrazi* $\frac{X_i(S)}{y_i(S)}$, $\frac{W_i(S)}{v_i(S)}$, $\frac{\Theta_i(S)}{\phi_i(S)}$ *su racionalne funkcije samo od* s_1 *i* s_2.

4.6 Metod pregrađivanja za težinske MP inverze polinomijalnih matrica sa više promenljivih

U ovom poglavlju opisaćemo algoritam za izračinavanje težinskih MP inverza polinomijalnih matrica sa više promenljivih. Ovaj algoritam je generalizacija metoda za izračunavanje težinskih MP inverza za konstantne matrice, koji je dat Algoritmom 3.4.2 a prvi put je opisan u [149]. Svi algoritmi o kojima će biti reči su implementirani u simboličkom programskom paketu MATHEMATICA.

U ovom poglavlju su izloženi naši originalni rezultati preuzeti iz radova [104, 141].

4.6.1 Racionalne matrice

Neka je $A(s_1, \ldots, s_p)$ kompleksna racionalna matrica. Da bi pojednostavili notaciju, uvešćemo nove oznake, slično kao u prethodnom poglavlju. Označimo nove promenljive sa $s_{2p+1-i} = \overline{s_i}$ i vektor svih promenljivih $s_1, \ldots, s_{2p}$ sa $S = (s_1, \ldots, s_{2p})$. Pored toga, definišimo operator konjugovanja $\overline{I}$ na indeksnoj $2p$-torci $I = (i_1, \ldots, i_{2p})$ kao $\overline{I} = (i_{2p}, i_{2p-1}, \ldots, i_1)$ i na isti način na $2p$-torci promenljivih $S = (s_1, \ldots, s_{2p})$ kao $\overline{S} = (s_{2p}, s_{2p-1}, \ldots, s_1)$. Ova definicija je u saglasnosti sa novim promenljivama.

Teorema 3.4.3 i Lema 3.4.4 takođe važe za racionalne matrice sa više promenljivih. Na taj način, Algoritam 3.4.2 i Algoritam 3.4.3 mogu da se direktno primene na racionalne matrice sa više promenljivih.

Podsetimo se da je $A_i(S)$ submatrica matrice $A(S)$ koja sadrži njenih prvih i kolona a da je $a_i(S)$ i-ta kolona matrice $A(S)$,

$$A_i(S) = [A_{i-1}(S) \mid a_i(S)]\,, i = 2, \ldots, n, \quad A_1(S) = a_1(S). \tag{4.21}$$

Kao i u prethodnom poglavlju, u implementaciji koristimo funkciju `Together` programskog paketa MATHEMATICA da bi omogućili uprošćavanje racionalnih izraza (ova funkcija grupiše racionalne sabirke i skraćuje zajedničke činioce u imeniocu i brojiocu).

4.6.2 Polinomijalne matrice

Pretpostavimo sada da je $A(S) \in \mathbb{C}[S]^{m\times n}$ polinomijalna matrica sa više promenljivih. Nju možemo da predstavimo u sledećem polinomijalnom obliku

$$A(S) = \sum_{i_1=0}^{d_1} \cdots \sum_{i_{2p}=0}^{d_{2p}} A_{i_1,\ldots,i_{2p}} s_1^{i_1} \cdots s_{2p}^{i_{2p}} = \sum_{I=0}^{Q} A_I S^I, \tag{4.22}$$

gde su $I = (i_1, \ldots, i_{2p})$, $A_I = A_{i_1,\ldots,i_{2p}}$ konstantne $m \times n$ matrice, $S^I = s_1^{i_1} s_2^{i_2} \cdots s_{2p}^{i_{2p}}$, $Q = (d_1, \ldots, d_{2p}) = \deg A(S)$. Ovde je $d_i = \deg_{s_i} A(S)$.

Ako sa $\overline{J}$ označimo $\overline{J} = (j_{2p}, \ldots, j_1)$, gde je $J = (j_1, \ldots, j_{2p})$ tada može lako da se proveri da važi $A^*(S) = \sum_{J=0}^{\overline{Q}} A^*_{\overline{J}} S^J$.

Modifikaciju Algoritma 3.4.2 za polinomijalne matrice $A(S)$ daje Teorema 4.6.1. U cilju pojednostavljenja relacija, u dokazu Teoreme 4.6.1 nećemo koristiti eksplicitan polinomijalan oblik pomoćnih matrica i skalara kao što smo to činili u prethodnom poglavlju. Napomenimo da su ovakvi izrazi izvedeni u našem radu [141] za slučaj polinomijalne matrice jedne promenljive.

Teorema 4.6.1. *Posmatrajmo matricu $A(S) \in \mathbb{C}[S]^{m\times n}$ u obliku (4.22) i pozitivno definitne Hermitske matrice $M(S) \in \mathbb{C}(S)^{m\times m}$ i $N(S) \in \mathbb{C}(S)^{n\times n}$. Pretpostavimo da je glavnodijagonalna podmatrica $N_i(S) \in \mathbb{C}(S)^{i\times i}$ matrice $N(S)$ pregrađena kao u (3.42). Tada je težinski MP inverz $A^{\dagger}_{MN_i}(S) \in \mathbb{C}^{i\times m}[S]$ koji odgovara prvim i kolonama matrice $A(S)$ oblika*

$$X_i(S) = A^{\dagger}_{MN_i}(S) = \frac{Z_i(S)}{Y_i(S)}, \; i = 1, \ldots, n, \tag{4.23}$$

gde $Z_i(S) \in \mathbb{C}^{m\times i}[S]$ i $Y_i(S) \in \mathbb{C}[S]$ može da se izračuna iz $Z_{i-1}(S)$, $Y_{i-1}(S)$, $A_{i-1}(S)$ i $a_i(S)$ koristeći egzaktne rekurentne relacije.

Dokaz. Teoremu ćemo dokazati matematičkom indukcijom. U slučaju $i = 1$ egzaktna relacija za $Z_1(S)$ i $Y_1(S)$ može da se izvede iz (3.43):

$$a_1(S) = A_1(S) = 0 \Rightarrow Z_1(S) = 0, \quad Y_1(S) = 1$$
$$a_1(S) = A_1(S) \neq 0 \Rightarrow Z_1(S) = a_1^*(S)M(S), \quad Y_1(S) = a_1^*(S)M(S)a_1(S)$$

Razmotrimo sada induktivni korak. Na osnovu induktivne hipoteze možemo da napišemo $X_{i-1}(S) = \frac{Z_{i-1}(S)}{Y_{i-1}(S)}$. Tada $X_i(S)$ može da se izračuna koristeći korake 3-6 Algoritma 3.4.2. Iz koraka 3 i 4 Algoritma 3.4.2 imamo

$$d_i(S) = X_{i-1}(S)a_i(S) = \frac{Z_{i-1}(S)a_i(S)}{Y_{i-1}(S)} = \frac{D_i(S)}{Y_{i-1}(S)}$$
$$c_i(S) = a_i(S) - A_{i-1}(S)d_i(S) = \frac{a_i(S)Y_{i-1}(S) - A_{i-1}(S)D_i(S)}{Y_{i-1}(S)} = \frac{C_i(S)}{Y_{i-1}(S)}.$$

Ako je $C_i(S) \neq 0$, saglasno koraku 5 Algoritma 3.4.2 imamo

$$b_i^*(S) = \frac{\frac{C_i^*(S)}{Y_{i-1}^*(S)}M(S)}{\frac{C_i^*(S)}{Y_{i-1}^*(S)}M(S)\frac{C_i(S)}{Y_{i-1}(S)}} = \frac{Y_{i-1}(S)C_i^*(S)M(S)}{C_i^*(S)M(S)C_i(S)} = \frac{V_i(S)}{W_i(S)}$$

U suprotnom, prvo treba da odredimo izraz $\delta_i(S)$. Iz (3.48) dobijamo

$$\begin{aligned}\delta_i(S) &= n_{ii}(S) + \frac{D_i^*(S)}{Y_{i-1}^*(S)}N_{i-1}(S)\frac{D_i(S)}{Y_{i-1}(S)} \\ &\quad - \left(\frac{D_i^*(S)}{Y_{i-1}^*(S)}l_i(S) + l_i^*(S)\frac{D_i(S)}{Y_{i-1}(S)}\right) - l_i^*(S)\frac{\phi_i(S)}{\psi_i(S)}.\end{aligned} \tag{4.24}$$

Sada koristimo induktivnu hipotezu zajedno sa pomoćnom polinomijalnom matricom $\phi_i(S) \in \mathbb{C}[S]^{(i-1)\times 1}$ i polinomom $\psi_i(S)$ koji su definisani sa

$$\begin{aligned}&\left(I - X_{i-1}(S)A_{i-1}(S)\right)N_{i-1}^{-1}(S)l_i(S) \\ &= \frac{Y_{i-1}(S)I - Z_{i-1}(S)A_{i-1}(S)}{Y_{i-1}(S)} \cdot \frac{\widetilde{N}_{i-1}(S)}{\breve{N}_{i-1}(S)} \cdot l_i(S) \\ &= \frac{Y_{i-1}(S)\widetilde{N}_{i-1}(S)l_i(S) - Z_{i-1}(S)A_{i-1}(S)\widetilde{N}_{i-1}(S)l_i(S)}{Y_{i-1}(S)\breve{N}_{i-1}(S)} = \frac{\phi_i(S)}{\psi_i(S)}.\end{aligned} \tag{4.25}$$

Takođe, koristimo $N_{i-1}^{-1}(S) = \frac{\widetilde{N}_i(S)}{\breve{N}_i(S)}$, gde su $\widetilde{N}_i(S) \in \mathbb{C}[S]^{(i-1)\times(i-1)}$ i $\breve{N}_i(S) \in \mathbb{C}[S]$ definisani narednom teoremom. Grupišući sabirke sa istim imeniocem u (4.24) možemo $\delta_i(S)$ da zapišemo u obliku

$$\delta_i(S) = \frac{\widetilde{\Delta}_i(S)}{\breve{\Delta}_i(S)}$$

gde je

$$\begin{aligned}\widetilde{\Delta}_i(S) &= n_{ii}(S)\breve{N}_{i-1}(S)Y_{i-1}^*(S)Y_{i-1}(S) + \breve{N}_{i-1}(S)D_i^*(S)N_{i-1}(S)D_i(S) \\ &\quad - \left(Y_{i-1}(S)D_i^*(S)l_i(S) + Y_{i-1}^*(S)D_i(S)l_i^*(S)\right)\breve{N}_{i-1}(S) - l_i^*(S)\phi_i(S)Y_{i-1}^*(S) \\ \breve{\Delta}_i(S) &= Y_{i-1}^*(S)Y_{i-1}(S)\breve{N}_{i-1}(S).\end{aligned}$$

Sada primenjujemo korak 5 Algoritma 3.4.2 za slučaj $C_i(S) = 0$ i izračunavamo $b_i^*(S)$ kao

$$\begin{aligned}b_i^*(S) &= \frac{\widetilde{\Delta}_i(S)}{\breve{\Delta}_i(S)}\left(\frac{D_i^*(S)}{Y_{i-1}^*(S)}N_{i-1}(S) - l_i(S)^*\right)\frac{Z_{i-1}(S)}{Y_{i-1}(S)} \\ &= \frac{\widetilde{N}_{i-1}(S)\left(D_i^*(S)N_{i-1}(S) - Y_{i-1}^*(S)l_i^*(S)\right)Z_{i-1}(S)}{\breve{\Delta}_i(S)} = \frac{V_i(S)}{W_i(S)}.\end{aligned}$$

Zapišimo sada izraz (3.44) na sledeći način

$$\begin{aligned}
X_i(S) &= \begin{bmatrix} \frac{Z_{i-1}(S)}{Y_{i-1}(S)} - \left(\frac{D_i(S)}{Y_{i-1}(S)} + \frac{\phi_i(S)}{\psi_i(S)}\right)\frac{V_i(S)}{W_i(S)} \\ \\ \frac{V_i(S)}{W_i(S)} \end{bmatrix} \\
&= \frac{1}{W_i(S)\psi_i(S)} \begin{bmatrix} W_i(S)\breve{N}_{i-1}(S)Z_{i-1}(S) - \left(D_i(S)\breve{N}_{i-1}(S) + \phi_i(S)\right) V_i(S) \\ \breve{N}_{i-1}(S)Y_{i-1}(S)V_i(S) \end{bmatrix}.
\end{aligned}$$

Iz poslednjeg izraza je očigledno da važi

$$\begin{aligned}
Z_i(S) &= \begin{bmatrix} W_i(S)\breve{N}_{i-1}(S)Z_{i-1}(S) - \left(D_i(S)\breve{N}_{i-1}(S) + \phi_i(S)\right) V_i(S) \\ \breve{N}_{i-1}(S)Y_{i-1}(S)V_i(S) \end{bmatrix} = \begin{bmatrix} \Theta_i(S) \\ \Psi_i(S) \end{bmatrix} \\
Y_i(S) &= W_i(S)\psi_i(S).
\end{aligned}$$

Ovim je dokaz teoreme završen. □

Teorema 4.6.2. *Neka je glavnodijagonalna podmatrica $N_i(S)$ Hermitske, pozitivno definitne matrice $N(S) \in \mathbb{C}[s]^{n\times n}$ pregrađena kao u (3.42). Tada je inverz $N_i^{-1}(S)$ oblika*

$$N_i^{-1}(S) = \frac{\widetilde{N}_i(S)}{\breve{N}_i(S)} = \frac{1}{\breve{N}_i(S)} \begin{bmatrix} E_{i-1}(S) & F_i(S) \\ F_i^*(S) & H_{ii}(S) \end{bmatrix}$$

gde $E_{i-1}(S) \in \mathbb{C}^{(i-1)\times(i-1)}$, $F_i^*(S) \in \mathbb{C}^{(i-1)\times 1}$ i polinom $H_{ii}(S) \in \mathbb{C}[s]$ mogu da se izračunaju iz $N_{i-1}(S)$, $l(S)$, $n_{ii}(S)$, $\widetilde{N}_{i-1}(S)$ i $\breve{N}_{i-1}(S)$ koristeći egzaktne rekurentne relacije.

Dokaz. Kao u dokazu prethodne teoreme koristićemo matematičku indukciju i Lemu 3.4.4 (Algoritam 3.4.3). Slučaj $i = 1$ je i ovde trivijalan, tako da imamo

$$\widetilde{N}_1(S) = 1, \quad \breve{N}_1(S) = n_{11}(S).$$

Dokažimo sada induktivni korak. Pretpostavimo da je $N_{i-1}^{-1}(S) = \frac{\widetilde{N}_{i-1}(S)}{\breve{N}_{i-1}(S)}$. Iz relacije (3.50) imamo

$$\begin{aligned}
\frac{1}{H_{ii}(S)} &= n_{ii}(S) - l_i^*(S)\frac{\widetilde{N}_{i-1}(S)}{\breve{N}_{i-1}(S)}l_i(S) \qquad (4.26) \\
&= \frac{\breve{N}_{i-1}(S)n_{ii}(S) - l_i^*(S)\widetilde{N}_{i-1}(S)l_i(S)}{\breve{N}_{i-1}(S)} = \frac{\breve{H}_i(S)}{\breve{N}_{i-1}(S)}.
\end{aligned}$$

Prema tome, možemo da napišemo $H_{ii}(S) = \frac{\breve{N}_{i-1}(S)}{\breve{H}_i(S)}$. Koristeći relaciju (3.51) možemo da $f_i(S)$ predstavimo na sledeći način

$$f_i(S) = -\frac{\widetilde{N}_{i-1}(S)}{\breve{H}_i(S)} \cdot l_i^*(S) \cdot \frac{\widetilde{N}_{i-1}(S)}{\breve{N}_{i-1}(S)} = -\frac{l_i^*(S)\widetilde{N}_{i-1}(S)}{\breve{H}_i(S)} = \frac{\widetilde{F}_i(S)}{\breve{H}_i(S)}.$$

Šta više, koristeći činjenicu da je $\widetilde{N}_{i-1}(S)$ Hermitska i pozitivno definitna matrica, zaključujemo da je $\widetilde{F}_i^*(S) = \widetilde{N}_{i-1}(S)l_i(S)$ odakle dalje sledi

$$f_i^*(S) = \frac{\widetilde{F}_i^*(S)}{\breve{H}_i^*(S)} = \frac{\widetilde{N}_{i-1}(S)l_i(S)}{\breve{H}_i(S)}.$$

Takođe, koristimo da je $\breve{H}_i(S) = \breve{H}_i^*(S)$, što može lako da se dokaže iz (4.26). Iz (3.52) zaključujemo

$$\begin{aligned} E_{i-1} &= \frac{\widetilde{N}_{i-1}(S)}{\breve{N}_{i-1}(S)} + \frac{\breve{H}_i(S)}{\breve{N}_{i-1}(S)} \frac{\widetilde{F}_i(S)}{\breve{H}_i(S)} \frac{\widetilde{F}_i^*(S)}{\breve{H}_i(S)} \\ &= \frac{\widetilde{N}_{i-1}(S) - \widetilde{F}_i(S)\widetilde{F}_i^*(S)}{\breve{N}_{i-1}(S)\breve{H}_i(S)} = \frac{\widetilde{E}_{i-1}(S)}{\breve{N}_{i-1}(S)\breve{H}_i(S)}. \end{aligned}$$

Konačno, $N_i^{-1}(S)$ možemo da predstavimo u sledećem matričnom obliku

$$\begin{aligned} N_i^{-1}(S) &= \begin{bmatrix} \frac{\widetilde{E}_{i-1}(S)}{\breve{N}_{i-1}(S)\breve{H}_i(S)} & \frac{\widetilde{F}_i(S)}{\breve{H}_i(S)} \\ \\ \frac{\widetilde{F}_i^*(S)}{\breve{H}_i(S)} & \frac{\breve{N}_{i-1}(S)}{\breve{H}_i(S)} \end{bmatrix} \\ &= \frac{1}{\breve{H}_i(S)\breve{N}_{i-1}(S)} \begin{bmatrix} \widetilde{E}_{i-1}(S) & \breve{N}_{i-1}(S)\widetilde{F}_i(S) \\ \breve{N}_{i-1}(S)\widetilde{F}_i^*(S) & \breve{N}_{i-1}(S)^2 \end{bmatrix} = \frac{\widetilde{N}_i(S)}{\breve{N}_i(S)}. \end{aligned}$$

Ovim je dokaz teoreme završen. □

Sada je lako da se konstruišu odgovarajući algoritmi bazirani na Teoremi 4.6.1 i Teoremi 4.6.2.

4.6.3 Retke strukture i modifikacija za retke matrice

U praksi se često radi sa matricama $A(S)$ koje imaju relativno mali broj nenultih koeficijenata. U tom slučaju, zbog velikog broja suvišnih operacija, algoritmi bazirani na Teoremi 4.6.1 i Teoremi 4.6.2 kao i Algoritam 4.5.1 nisu efikasni. Da bi izbegli taj problem konstruisaćemo dve pogodne retke strukture koje reprezentuju polinomijalnu matricu $A(S)$ i odgovarajući efikasan algoritam za izračunavanje $A^{\dagger}_{MN}(S)$. Prva retka struktura je označena sa **Eff** a njeno poboljšanje sa **Eff**$'$, dok je druga struktura označena sa **Ef**.

Vredi napomenuti da svi rezultati u ovom odeljku mogu da se primene na Algoritam 4.5.1 iz prethodnog poglavlja. Ovo je urađeno u našem radu [101] i korišćeno je u implementaciji Algoritma 4.5.1.

Osnovna ideja pri razmatranju prve retke strukture je da se koriste samo nenulti matrica koeficijenti $A_I = A_{i_1,\dots,i_{2p}} \neq 0$ polinomijalne matrice $A(S)$ dati u obliku (4.22).

Definicija 4.6.1. *Definisaćemo sledeću retku (efikasnu) strukturu polinomijalne matrice $A(S)$ sa*

$$\mathbf{Eff}_A = \{(J, A_J) \,|\, A_J \neq 0,\ 0 \leq J \leq \deg A(S)\}. \tag{4.27}$$

Takođe definišimo skup indeksa ove retke strukture kao

$$\mathrm{Ind}_A = \{J \mid A_J \neq 0,\ 0 \leq J \leq \deg A(S)\}\,. \tag{4.28}$$

Definišimo operacije $+$, $-$, $\cdot$ *i* $*$ *na retkim strukturama kao*

$$\begin{aligned} &\mathbf{Eff}_A + \mathbf{Eff}_B = \mathbf{Eff}_{A+B},\ \mathbf{Eff}_A - \mathbf{Eff}_B = \mathbf{Eff}_{A-B},\\ &\mathbf{Eff}_A \cdot \mathbf{Eff}_B = \mathbf{Eff}_{A\cdot B},\quad \mathbf{Eff}^*_A = \mathbf{Eff}_{A^*}. \end{aligned} \tag{4.29}$$

Označimo sa $e_A = |\mathbf{Eff}_A| = |\mathrm{Ind}_A|$ *veličinu strukture* $\mathbf{Eff}_A$.

Očigledno važi

$$A(S)\cdot B(S) = \sum_{\substack{I\in \mathrm{Ind}_A\\ J\in \mathrm{Ind}_B}} A_I B_J S^{I+J},$$

gde je

$$S^{I+J} = s_1^{i_1+j_1}\cdots s_{2p}^{i_{2p}+j_{2p}}.$$

Ako je $C(S) = A(S)B(S)$ tada su elementi matrice $\mathbf{Eff}_C$ parovi (K, C_K) gde je C_K definisano kao sledeća suma proizvoda matrica

$$C_K = \sum_{\substack{I\in \mathrm{Ind}_A\\ K-I\in \mathrm{Ind}_B}} A_I B_{K-I} \tag{4.30}$$

pri čemu je $C_K \neq 0$. Prema tome, važi $e_C \leq e_A + e_B$ i $\mathbf{Eff}_C = \mathbf{Eff}_A \cdot \mathbf{Eff}_B$ može da se izračuna u vremenu $O(e_A \cdot e_B)$.

Slično važi za izračunavanje zbira $C(S) = A(S) + B(S)$. Elementi matrice $\mathbf{Eff}_C$ su parovi (K, C_K) gde su vrednosti C_K definisane sa

$$C_K = \begin{cases} A_K, & A_K \neq 0, B_K = 0\\ B_K, & B_K \neq 0, A_K = 0\\ A_K + B_K, & A_K \neq 0, B_K \neq 0 \end{cases} \tag{4.31}$$

i zadovoljavaju $C_K \neq 0$. Kao i u prethodnom slučaju, možemo da zaključimo da $e_C \leq \max\{e_A, e_B\}$ i $\mathbf{Eff}_C$ može da se izračuna vremenu $\mathcal{O}(\max\{e_A, e_B\})$.

Skupovi indeksa koji odgovaraju sabiranju i množenju matrica su

$$\mathrm{Ind}_{A+B} = \mathrm{Ind}_A \cup \mathrm{Ind}_B,\quad \mathrm{Ind}_{AB} = \mathrm{Ind}_A + \mathrm{Ind}_B$$

S obzirom na (4.29), $\mathbf{Eff}^*_A = \{(I, A^*_I) \mid (I, A_I) \in \mathbf{Eff}_A\}$ izračunavamo u vremenu $\mathcal{O}(e_A)$.

Obično su koeficijent matrice A_I u polinomijalnoj reprezentaciji (4.22), tj. u retkoj reprezentaciji (4.27) retke. Koristeći ovu činjenicu možemo značajno poboljšati našu retku strukturu **Eff** koristeći pogodnu strukturu za ove konstantne koeficijent matrice.

Definicija 4.6.2. *Za konstantnu matricu* $A = [a_{ij}] \in \mathbb{C}^{m \times n}$*, definišimo sledeću retku strukturu*

$$\mathbf{Sp}_A = \{(i, j, a_{ij}) \mid a_{ij} \neq 0\} \tag{4.32}$$

Označimo sa $s_A = |\mathbf{Sp}_A|$ *veličinu strukture* $\mathbf{Sp}_A$*.*

Slično kao u slučaju $\mathbf{Eff}_A$, definisaćemo elementarne operacije na ovim retkim strukturama

$$\begin{aligned}
\mathbf{Sp}_A + \mathbf{Sp}_B &= \{(i, j, a_{ij} + b_{ij}) \mid (i, j, a_{ij}) \in \mathbf{Sp}_A \vee (i, j, b_{ij}) \in \mathbf{Sp}_B,\ a_{ij} + b_{ij} \neq 0\} \\
\mathbf{Sp}_A \cdot \mathbf{Sp}_B &= \{(i, j, c_{ij}) \mid c_{ij} = \sum a_{ik} b_{kj} \neq 0, (i, k, a_{ik}) \in \mathbf{Sp}_A \wedge (k, j, b_{kj}) \in \mathbf{Sp}_B\} \\
\mathbf{Sp}^*_A &= \{(j, i, a^*_{ij}) \mid (i, j, a_{ij}) \in \mathbf{Sp}_A\}
\end{aligned}$$

Na taj način, imamo sledeće poboljšanje strukture $\mathbf{Eff}$:

$$\begin{aligned}
\mathbf{Eff}'_A &= \left\{(J, \mathbf{Sp}_{A_J}) \,|\, A_J \neq 0,\ 0 \leq J \leq \deg A(S)\right\} \\
&= \left\{(J, \{i, j, (A_J)_{ij} | (A_J)_{ij} \neq 0\}) \mid A_J \neq 0,\ 0 \leq J \leq \deg A(S)\right\}.
\end{aligned} \tag{4.33}$$

Može se videti da je složenost izračunavanja $\mathbf{Sp}_A + \mathbf{Sp}_B$ jednaka $\mathcal{O}(s_A + s_B)$ a za $\mathbf{Sp}^*_A$ je $\mathcal{O}(s_A)$. U slučaju množenja složenost izračunavanja zavisi od konkretne implementacije. Pretpostavimo da je $A \in \mathbb{C}^{m \times n}$ i $B \in \mathbb{C}^{n \times p}$. Ako se trojke urede leksikografski u $\mathbf{Sp}_A$ i u $\mathbf{Sp}_B$ tada za svako $(i, k, A_{ik}) \in \mathbf{Sp}_A$ treba da nađemo sve $(k, j, B_{kj}) \in \mathbf{Sp}_B$, tj. sve trojke u $\mathbf{Sp}_B$ koje počinju sa k. Ako taj broj obeležimo sa $s_B^{(k)}$, tj. ako je

$$s_B^{(k)} = |\{(k, j, b_{kj}) \in \mathbf{Sp}_B \mid j = 1, \ldots, p\}|$$

tada je složenost množenja $\mathbf{Sp}_A \cdot \mathbf{Sp}_B$ data sa

$$\mathcal{O}\left(\sum_{(i,k,a_{ik}) \in \mathbf{Sp}_A} s_B^{(k)} + m \cdot p\right). \tag{4.34}$$

Zadnji sabirak u (4.34) se javlja zbog činjenice da je potrebno da se konstruiše retka struktura $\mathbf{Sp}_C$ za matricu $C = AB \in \mathbb{C}^{m \times p}$.

Retka struktura $\mathbf{Sp}$ je već implementirana u programskom paketu MATHEMATICA kao struktura `SparseArray`, [157, 158]. Oba izraza

`SparseArray`$[\{i_1, j_1\} -> v_1, \{i_2, j_2\} -> v_2, \ldots,]$

`SparseArray`$[\{\{i_1, j_1\}, \{i_2, j_2\}, \ldots\} -> \{v_1, v_2, \ldots\}]$

reprezentuju retku strukturu sa elementima u pozicijama $\{i_k, j_k\}$ koji imaju vrednosti v_k. Operacije sa retkim matricama su potpuno ekvivalentne operacijama sa gustim matricama [157, 158]: `Plus(+)` za sabiranje matrica, `Dot (.)` za množenje matrica, `Times (*)` za množenje skalarom, itd.

Prema tome, u našoj implementaciji imamo da je

$$\mathbf{Eff}'_A = \{(J, \texttt{SparseArray}[A_J]) \,|\, A_J \neq 0,\ 0 \leq J \leq \deg A(S)\}.$$

Činjenica da su osnovne operacije iste i za guste i za retke matrice omogućuje da se koriste iste procedure za osnovne operacije na **Eff** u slučaju kada je **Sp** ugrađeno u **Eff** i kada to nije. U proceduralnim programskim jezicima na početku algoritma moramo da odlučimo da li koristimo **Sp** ili ne, zavisno od strukture ulaznih matrica $A(S)$, $M(S)$ i $N(S)$. Na sličan način, moguće je promeniti izbor jedne od ove dve varijante strukture **Eff** tokom implementacije algoritma.

U drugoj retkoj strukturi matricu $A(S)$ predstavljamo u obliku $A(S) = [a_{ij}(S)]$, gde su $a_{ij}(S)$ skalarni polinomi i konstruišemo retku strukturu $\mathbf{Eff}_{a_{ij}}$ za svako $a_{ij}(S)$. Retka struktura $\mathbf{Eff}_a$ za skalarni polinom $a(S) = \sum_{I=1}^{\deg a(S)} a_I S^I$ se definiše slično kao i u slučaju matrice (4.27):

$$\mathbf{Eff}_a = \{(J, a_J) \mid a_J \neq 0,\ 1 \leq J \leq \deg a(S)\}.$$

Označimo ovakvu retku reprezentaciju sa **Ef**. Drugim rečima važi $\mathbf{Ef}_A = [\mathbf{Eff}_{a_{ij}}]$. Ako koristimo oznake $\mathbf{ef}_A = \sum_{i=1}^{m} \sum_{j=1}^{n} e_{a_{ij}}$, složenost operacije sabiranja **Ef** struktura je

$$\mathcal{O}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} e_{a_{ij}} + e_{b_{ij}}\right) = O(\mathbf{ef}_A + \mathbf{ef}_B).$$

Označimo sada $\mathbf{row}(B,k) = \sum_{j=1}^{p} e_{b_{kj}}$ i $\mathbf{col}(A,k) = \sum_{i=1}^{m} e_{a_{ik}}$. Složenost množenja matrica A i B predstavljenih preko efektivnih struktura **ef** jednaka je

$$\begin{aligned}\mathcal{O}\left(\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{j=1}^{p} e_{a_{ik}} e_{b_{kj}}\right) &= \mathcal{O}\left(\sum_{k=1}^{n}\sum_{i=1}^{m} e_{a_{ik}} \mathbf{row}(A,k)\right)\\ &= \mathcal{O}\left(\sum_{k=1}^{n} \mathbf{col}(A,k)\mathbf{row}(B,k)\right)\end{aligned}$$

Polinomi u programskom paketu MATHEMATICA se predstavljaju u internom obliku koristeći neznatno modifikovanu **Ef** retku strukturu. Na primer, polinom sa dve promenljive $p(s_1, s_2) = 4s_1^9 s_2^{10} + s_2^3 + s_1^2 s_2^2 + 3s_1^3 s_2 + s_2 + 2s_1^2 + 3s_1 + 10$ je u programskom paketu MATHEMATICA predstavljen sledećom internom reprezentacijom:

```
Plus[
10,
Times[3, s1],
Times[2, Power[s1, 2]],
s2,
Times[3, Power[s1, 3], s2],
Times[Power[s1, 2], Power[s2, 2]],
Power[s2, 3],
Times[4, Power[s1, 9], Power[s2, 10]]
].
```

Poslednji izraz je dobijen koristeći funkciju `FullForm[E]` programskog paketa MATHEMATICA koja vraća internu reprezentaciju izraza `E` [157, 158]. Ova interna reprezentacija polinoma $p(S)$, na najvišem nivou je lista dužine $\mathbf{ef}_p$ sa zaglavljem `Plus`. Svaki element ove liste sadrži eksponent i $J = (j_1, j_2)$ i vrednost p_J (vrednosti $j_1 = 0, 1$ i $j_2 = 0, 1$ nisu pokazane). Veličina cele strukture u memoriji je $\mathcal{O}(\mathbf{ef}_{p(s)})$. Prema tome možemo da koristimo ovu prirodnu polinomijalnu reprezentaciju u programskom paketu MATHEMATICA i da ugradimo elementarne operatore da bi implementirali efikasan metod pregrađivanja koristeći **Ef** strukturu. Složenost ovih ugrađenih operatora je ista kao i složenost odgovarajućih operatora definisanih za **Ef** strukturu.

Algoritam 4.6.1 i Algoritam 4.6.2 implementiraju efikasni metod pregrađivanja za izračunavanje težinskih MP inverza polinomijalnih matrica i metod za izračunavanje inverza Hermitskih, pozitivno definitnih polinomijalnih matrica. Oba algoritma su pogodna za retke matrice. Generalno, metodi mogu da se koriste sa obe, predhodno opisane retke strukture. Prema tome, obeležićemo *opštu retku strukturu* sa $\mathcal{E}$. Ona može da se zameni bilo sa **Eff**, bilo sa **Ef**. Takođe, sa **O** označićemo opštu efikasnu strukturu odgovarajuće nula matrice. Isti simbol će biti korišćen za efikasnu strukturu broja 0.

Algoritam 4.6.1 Efikasno izračunavanje inverza Hermitske, pozitivno definitne polinomijalne matrice sa više promenljivih

$\mathcal{E}_{\widetilde{N}_1} := \mathcal{E}_I$
$\mathcal{E}_{\breve{N}_1} := \mathcal{E}_{n_{11}}$
for $i := 2$ to n **do**
  $\mathcal{E}_{\breve{H}_i} := \mathcal{E}_{n_{ii}} \cdot \mathcal{E}_{\breve{N}_{i-1}} - \mathcal{E}^*_{l_i} \cdot \mathcal{E}_{\widetilde{N}_{i-1}} \cdot \mathcal{E}_{l_i}$
  $\mathcal{E}_{\widetilde{F}_i} := -\mathcal{E}_{\widetilde{N}_{i-1}} \cdot \mathcal{E}_{l_i}$
  $\mathcal{E}_{\widetilde{E}_i} := \mathcal{E}_{\widetilde{N}_{i-1}} - \mathcal{E}_{F_i} \cdot \mathcal{E}^*_{F_i}$
  $\mathcal{E}_{\breve{N}_i} := \mathcal{E}_{\breve{N}_{i-1}} \cdot \mathcal{E}_{\breve{H}_i}$
  Formiraj $\mathcal{E}_{\widetilde{N}_i}$ koristeći

$$\widetilde{N}_i(S) = \begin{bmatrix} \widetilde{E}_{i-1}(S) & \breve{N}_{i-1}(S) \cdot \widetilde{F}_i(S) \\ \breve{N}_{i-1}(S) \cdot \widetilde{F}^*_i(S) & \breve{N}^2_{i-1}(S) \end{bmatrix}$$

end for
return $N_k^{-1}(S) = \frac{\widetilde{N}_k(S)}{\breve{N}_k(S)}$ za svaki $k = 1, \dots, n$.

4.6.4 Primeri

Algoritam 3.4.2, Algoritam 3.4.3, Algoritam 4.6.1 i Algoritam 4.6.2 su implementirani u programskom jeziku MATHEMATICA. Takođe je implementiran i algoritam za određivanje retke strukture **Eff**. Funkcije `WPolyEf` i `WPolyEff` implementiraju Algoritam 4.6.2 koristeći respektivno retke strukture **Ef** i **Eff**. Sve osnovne operacije za retku strukturu **Eff** (funkcije `Add`, `Sub`, `Muls`, `Mul` i `TE` koje odgovaraju sabiranju, oduzimanju, množenju skalarom, množenju i operaciji konjugovanja i transponovanja respektivno) su implementirane.

Algoritam 4.6.2 Efikasan metod za izračunavanje težinskog MP inverza retke matrice

Input: Efikasne strukture $\mathcal{E}_A, \mathcal{E}_M$ i $\mathcal{E}_N$ matrica $A(S) \in \mathbb{C}^{m\times n}$, $M(S) \in \mathbb{C}^{m\times m}$, $N(S) \in \mathbb{C}^{n\times n}$

Izračunaj inverze matrica $N_k^{-1}(S) = \frac{\widetilde{N}_k(S)}{\check{N}_k(S)}$ za svako $k = 1, \dots, n$ koristeći Algoritam 4.6.1.
if $\mathcal{E}_{a_1} = \mathbf{O}$ **then**
  $\mathcal{E}_{Z_1} := \mathbf{O}$
  $\mathcal{E}_{Y_1} := \mathcal{E}_1$ ($\mathcal{E}_1$ je retka struktura broja 1)
else
  $\mathcal{E}_{Z_1} := \mathcal{E}_{a_1}^* \cdot \mathcal{E}_M$
  $\mathcal{E}_{Y_1} := \mathcal{E}_{a_1}^* \cdot \mathcal{E}_M \cdot \mathcal{E}_{a_1}$
end if
for $i := 2$ to n **do**
  $\mathcal{E}_{d_i} := \mathcal{E}_{Z_{i-1}} \cdot \mathcal{E}_{a_i}$
  $\mathcal{E}_{c_i} := \mathcal{E}_{a_i} \cdot \mathcal{E}_{Y_{i-1}} - \mathcal{E}_{A_{i-1}} \cdot \mathcal{E}_{d_i}$
  if $\mathcal{E}_{c_i} \neq \mathbf{O}$ **then**
    $\mathcal{E}_{V_i} := \mathcal{E}_{Y_{i-1}} \cdot \mathcal{E}_{c_i}^* \cdot \mathcal{E}_M$
    $\mathcal{E}_{W_i} := \mathcal{E}_{c_i}^* \cdot \mathcal{E}_M \cdot \mathcal{E}_{c_i}$
  else
    $\mathcal{E}_{\varphi_i} := (\mathcal{E}_{Y_{i-1}} \cdot \mathcal{E}_I - \mathcal{E}_{Z_{i-1}} \cdot \mathcal{E}_{A_{i-1}}) \cdot \mathcal{E}_{\widetilde{N}_{i-1}} \cdot \mathcal{E}_{l_i}$
    $\mathcal{E}_{\psi_i} := \mathcal{E}_{Y_{i-1}} \cdot \mathcal{E}_{\check{N}_{i-1}}$
    $\mathcal{E}_{V_i} := \mathcal{E}_{\check{\Delta}_i} \cdot (\mathcal{E}_{d_i}^* \cdot \mathcal{E}_{N_{i-1}} - \mathcal{E}_{l_i}^* \cdot \mathcal{E}_{Y_{i-1}}) \cdot \mathcal{E}_{Z_{i-1}}$
    $\mathcal{E}_{W_i} := \mathcal{E}_{\widetilde{\Delta}_i} \cdot \mathcal{E}_{Y_{i-1}}^* \cdot \mathcal{E}_{Y_{i-1}}$
  end if
  if $\mathcal{E} = \mathbf{Eff}$ **then**

$$
\begin{aligned}
\mathbf{Eff}_{Z_i} := &\left\{ \left(j, \begin{bmatrix} (\Theta_i)_j \\ (\Psi_i)_j \end{bmatrix} \right) \mid (j, (\Theta_i)_j) \in \mathbf{Eff}_{\Theta_i}, (j, (\Psi_i)_j) \in \mathbf{Eff}_{\Psi_i} \right\} \\
&\cup \left\{ \left(j, \begin{bmatrix} (\Theta_i)_j \\ 0 \end{bmatrix} \right) \mid (j, (\Theta_i)_j) \in \mathbf{Eff}_{\Theta_i}, (\Psi_i)_j = 0 \right\} \\
&\cup \left\{ \left(j, \begin{bmatrix} 0 \\ (\Psi_i)_j \end{bmatrix} \right) \mid (\Theta_i)_j = 0, (j, (\Psi_i)_j) \in \mathbf{Eff}_{\Psi_i} \right\}
\end{aligned}
$$

  else
    $\mathbf{Ef}_{Z_i} := \begin{bmatrix} \mathbf{Ef}_{\Theta_i} \\ \mathbf{Ef}_{\Psi_i} \end{bmatrix}$
  end if
  $\mathcal{E}_{\Theta_i} := \mathcal{E}_{Z_{i-1}} \cdot \mathcal{E}_{\check{N}_{i-1}} \cdot \mathcal{E}_{W_i} - \mathcal{E}_{d_i} \cdot \mathcal{E}_{\check{N}_{i-1}} \cdot \mathcal{E}_{V_i} - \mathcal{E}_{\varphi_i} \mathcal{E}_{V_i}$
  $\mathcal{E}_{\Psi_i} := \mathcal{E}_{\psi_i} \cdot \mathcal{E}_{V_i}$
  $Z_i := \begin{bmatrix} \Theta_i \\ \Psi_i \end{bmatrix}$
  $\mathcal{E}_{Y_i} := \mathcal{E}_{\psi_i} \cdot \mathcal{E}_{W_i}$
  Nađi polinome $Z_i(S)$ i $Y_i(S)$ iz njihovih efikasnih struktura
  $X_i(S) := \frac{Z_i(S)}{Y_i(S)}$
  Skrati zajedničke činioce u brojiocu $Z_i(S)$ i imeniocu $Y_i(S)$ i preračunaj (ako je potrebno) efikasne strukture.
end for
return $A^{\dagger}_{M(S),N(S)}(S) := X_n(S)$

Primer 4.6.1. Odredimo težinski MP inverz sledeće polinomijalne matrice sa dve promenljive $A(x,y)$:

$$A(x,y)=\begin{bmatrix}1-3x & 5+9x-10y & 16+8x+2y\\ -7+9x-8y & 8+5x-y & 4+2x+3y\\ 7-x-8y & 16-2x-6y & -3-2x-4y\end{bmatrix}$$

U odnosu na sledeće matrice $M(x,y)$ i $N(x,y)$:

$$\begin{aligned}M(x,y) &= \begin{bmatrix}-20-x-\overline{x} & -8-7x-4\overline{x} & -2(8+3x+4\overline{x})\\ -8-4x-7\overline{x} & -20+7x+7\overline{x} & 2(5x-\overline{x})\\ -2(8+4x+3\overline{x}) & -2(x-5\overline{x}) & 7(-2+x+\overline{x})\end{bmatrix}\\ N(x,y) &= \begin{bmatrix}16+7x+7\overline{x} & 7-6x-2\overline{x} & 6-10x-3\overline{x}\\ 7-2x-6\overline{x} & -2(3+5x+5\overline{x}) & -2(6+4x+3\overline{x})\\ 6-3x-10\overline{x} & -2(6+3x+4\overline{x}) & -3(-6+x+\overline{x})\end{bmatrix}.\end{aligned}$$

Obe funkcije `WPolyEf` i `WPolyEff` vraćaju istu vrednost za težinski MP inverz

$$X(x,y)=A^{\dagger}_{M(x,y),N(x,y)}(x,y)=\left(60x^3-5yx^2-540x^2+51yx+779x-42y-435\right)^{-1}$$
$$\times\begin{bmatrix}-5x^2+51x-42 & -3x^2+8x-13 & -3x^2+33x-4\\ -30x^2+71x+15 & 42x^2-5yx-33x+y+15 & -18x^2-63x+10y+105\\ -2\left(10x^2-19x+12\right) & 2\left(18x^2-2yx-29x+2y+17\right) & -24x^2+yx+42x-2y-23\end{bmatrix}$$

Napomenimo da su stepeni međurezultata u Algoritmu 4.6.1 i Algoritmu 4.6.2 mnogo veći od stepena matrica A, M, N i X (maksimalni stepeni u ovom primeru su 874 i 122 za promenljive x i y respektivno). Ovo je razlog zbog koga su prikazani algoritmi za izračunavanje težinskih MP inverza za polinomijalne matrice veoma spori (vreme izvršenja funkcije `WPolyEff` za poslednji primer je 172.922 sekunde). Kao što će se videti u nastavku, kada su matrice A, M i N retke, odgovarajući međurezultati su takođe retke matrice. Prema tome, retke strukture uvedene u prethodnom odeljku poboljšavaju vreme izvršenja implementacije.

Algoritam 4.6.2 je testiran na nekoliko slučajno generisanih test primera. Testirane su varijante Algoritma 4.6.2 u kojima se koriste retke strukture **Ef** i **Eff**. Pritom su matrice $A(S)$, $M(S)$ i $N(S)$ su kompleksne polinomijalne matrice jedne promenljive s (tj. važi $S=(s,\overline{s})$).

Testiranje je sprovedeno za dve različite klase matrica: retke i guste. Rezultati su dati u sledećoj tabeli (kolona d se odnosi na stepen odgovarajućih matrica polinoma $A(S)$, $M(S)$ i $N(S)$):

m	n	d	Alg 4.6.2 sa **Ef**	Alg. 4.6.2 sa **Eff**
2	2	1	0.14	0.188
2	2	2	0.65	1.24
2	2	3	1.92	3.93
3	3	1	1.34	1.32
3	3	2	9.01	11.81
3	3	3	34.39	48.13
4	4	1	7.87	6.74
4	4	2	69.31	64.48
4	4	3	461.07	594.98
5	5	1	49.13	58.48
5	5	2	309.38	330.32

$sp_1(A(S)) = 0.9,\ sp_2(A(S)) = 0.9$

m	n	d	Alg 4.6.2 sa **Ef**	Alg. 4.6.2 sa **Eff**
2	2	1	0.06	0.89
2	2	2	0.25	0.46
2	2	3	0.60	1.23
3	3	1	0.47	0.68
3	3	2	4.60	7.18
3	3	3	14.89	24.65
4	4	1	6.10	6.18
4	4	2	34.95	39.68
4	4	3	256.31	299.61
5	5	1	30.85	39.43
5	5	2	246.32	283.12

$sp_1(A(S)) = 0.7,\ sp_2(A(S)) = 0.5$

m	n	d	Alg 4.6.2 sa **Ef**	Alg. 4.6.2 sa **Eff**
2	2	1	0.04	0.112
2	2	2	0.11	0.263
2	2	3	0.422	1.303
3	3	1	0.281	0.972
3	3	2	1.367	3.505
3	3	3	5.808	18.449
4	4	1	1.613	5.549
4	4	2	12.134	27.113
4	4	3	55.139	107.27
5	5	1	7.475	13.582
5	5	2	84.712	139.681

$sp_1(A(S)) = 1,\ sp_2(A(S)) = 0.2$

m	n	d	Alg 4.6.2 sa **Ef**	Alg. 4.6.2 sa **Eff**
2	2	1	0.032	0.105
2	2	2	0.069	0.190
2	2	3	0.187	0.713
3	3	1	0.185	0.675
3	3	2	0.628	2.944
3	3	3	1.031	3.275
4	4	1	0.987	4.344
4	4	2	6.087	25.263
4	4	3	27.466	176.581
5	5	1	3.294	15.853
5	5	2	42.159	171.416

$sp_1(A(S)) = 0.2,\ sp_2(A(S)) = 0.2$

Sva vremena izvršenja koja su prikazana su u sekundama a spars brojevi za matrice $M(S)$ i $N(S)$ su isti kao odgovarajući spars brojevi za $A(S)$. Svako vreme izvršenja je dobijeno usrednjavanjem vremena izvršenja za 10 različitih slučajno generisanih test primera. Testiranje je izvršeno na Intel Pentium 4 procesoru na 2.6 GHz i sa programskim paketom MATHEMATICA 5.2. Možemo napomenuti da Algoritam 4.6.2 sa strukturom **Ef** pokazuje bolja vremena na svim test primerima. Već smo spomenuli da je algoritam za formiranje retke strukture **Ef** već implementiran u programskom paketu MATHEMATICA. U implementaciji su korišćeni standardni ugrađeni operatori za manipulaciju sa matricama u strukturi **Ef**.

Prva tabela (kada je $sp_1(A(S)) = sp_2(A(S)) = 0.9$) odgovara gustim matricama. U tom slučaju, retke strukture nisu dovoljno efikasne jer postoji mnogo nenultih elemenata u svim matricama i nenula koeficijenata u polinomima. Ali može da se primeti značajno poboljšanje vremena izvršenja kada je primenjena struktura **Ef** u odnosu na slučaj kada je primenjena struktura **Eff**. Ova razlika uglavnom dolazi zbog činjenice da je struktura **Ef** implementirana koristeći ugrađene operacije u programski paket MATHEMATICA.

Drugi slučaj (kada je $sp_1(A(S)) = 0.7$ i $sp_2(A(S)) = 0.5$) se odnosi na retke matrice. Možemo da napomenemo da je vreme izvršenja značajno manje nego u prvom slučaju. Takođe, ovde struktura **Ef** daje manje vreme izvršenja nego struktura **Eff**.

U trećem i četvrtom slučaju (kada je $sp_1(A(S)) = 1$ i $sp_2(A(S)) = 0.2$, i $sp_1(A(S)) = sp_2(A(S)) = 0.2$, respektivno) radi se o matricama čiji su elementi veoma retki polinomi. Štaviše, u četvrtom slučaju se radi o matricama sa samo nekoliko nenultih elemenata. U četvrtom slučaju, najmanje srednje vreme izvršenja je dobijeno za sve razmatrane dimenzije i stepene matrica. Napomenimo da kada spars broj opada, srednje vreme izvršenja takođe opada (za konstantne dimenzije i stepen matrice). Ovo važi za obe retke strukture i potvrđuje teorijske rezultate o retkim strukturama **Ef** i **Eff** u praksi.

Takođe je razmotren prostiji slučaj: kada se podrazumeva da su sve ulazne matrice ($A(S)$, $M(S)$ i $N(S)$) i promenljive $s_1, \ldots, s_p$ realne. U tom slučaju imamo samo p promenljivih i operacija konjugovanja-transponovanja se svodi samo na transponovanje.

Algoritam 4.6.1 i Algoritam 4.6.2 ostaju isti, osim što možemo da izmenimo definiciju operatora konjugovanja-transponovanja (isto važi i za implementacije u programskom paketu MATHEMATICA). Ovaj slučaj je razmatran u našem radu [141]. Algoritam 4.6.1 i Algoritam 4.6.2 su efikasne verzije odgovarajućih Algoritama 3.1 i 3.2 u [141]. Ovde je vreme izvršenja algoritama značajno manje i inverzi imaju mnogo manje stepene. Rezultati dobijeni u ovom specijalnom slučaju prikazani su u sledećoj tabeli:

m	n	d	Alg 3.4.2	Alg 4.6.2 sa **Eff**	Alg. 4.6.2 sa **Ef**	Alg 3.1 sa [141]
3	3	1	0.32	0.23	0.10	0.94
3	3	2	0.69	0.57	0.20	1.32
3	3	3	0.82	1.17	0.43	1.84
3	3	4	1.19	2.15	0.73	2.38
4	3	1	0.76	1.26	0.14	1.29
4	3	2	1.29	0.65	0.31	2.12
4	3	3	2.14	1.32	0.59	2.42
4	3	4	2.84	2.26	1.01	2.93
5	5	1	3.48	1.45	1.01	3.56
5	5	2	5.90	4.54	2.92	4.92
5	5	3	9.18	8.79	6.82	8.27
5	5	4	12.15	15.87	10.85	10.34
6	6	1	7.98	2.65	2.17	8.16
6	6	2	12.93	8.20	7.31	11.32
6	6	3	21.76	18.29	13.53	19.42

$sp_1(A(S)) = 0.7$, $sp_2(A(S)) = 0.7$

Iz tabele može da se vidi da ovde u svim slučajevima struktura **Ef** daje bolje rezultate nego struktura **Eff** (obe uz korišćenje Algoritma 4.6.2). Oba efikasna algoritma su značajno bolja od Algoritma 3.4.2 (za racionalne matrice) i Algoritma 3.1 iz [141]. Za manje vrednosti d, Algoritam 3.4.2 je bolji od Algoritma 3.1 iz [141] zbog implementacionih detalja.

Svi izloženi rezultati dovode do istog zaključka: najbolji metod za izračunavanje težinskih MP inverza polinomijalnih matrica je Algoritam 4.6.2 sa retkom strukturom **Ef**.

4.7 Primene

Generalisani inverzi su veoma moćan alat i primenjuju se u mnogim granama matematike (takođe i u drugim naukama i u tehnici). Već smo videli neke primene generalisanih inverza za rešavanje matričnih jednačina (odeljak 3.1). U mnogim drugim disciplinama generalisani inverzi igraju značajnu ulogu, na primer: matematička statistika (regresija), izračunavanje polarne dekompozicije, teorija električnih kola, teorija automatskog upravljanja, filtriranje signala, diferencne jednačine, prepoznavanje slike, itd. Vredno je spomenuti da je glavna primena generalisanih inverza model linearne regresije, koji ćemo izložiti u sledećem odeljku. Napomenimo da je mnogo detaljnije razmatranje klasičnih primena generalisanih inverza dato u monografiji [10]. U ovom poglavlju ukratko ćemo izložiti dve primene generalisanih inverza: u matematičkoj statistici (linearna regresija) i teoriji automatskog upravljanja (projektovanje sistema sa povratnom vezom). Nadamo se da ova dva primera ilustruju snagu i mogućnosti primene generalisanih inverza.

4.7.1 Linearna regresija

U mnogim naukama veoma često postoji potreba određivanja direktne funkcionalne zavisnosti između dve ili više veličina. Oblast matematičke statistike koja se bavi utvrđivanjem i opisivanjem ovih zavisnosti naziva se *regresija*. Postoje dva moguća pristupa ovom problemu.

Kod prvog pristupa posmatra se uticaj obeležja (slučajnih veličina) $X_1, \dots, X_n$ na takođe slučajnu veličinu Y. Pri tom, treba odrediti funkciju $f(x_1, \dots, x_n)$ takvu da slučajna veličina $f(X_1, \dots, X_n)$ najbolje aproksimira Y. Kao mera odstupanja ovih dveju veličina, najčešće se koristi srednjekvadratno odstupanje, tj. uslov da je $E(Y - f(X))^2$ minimalno. Ovim tipom regresije nećemo se baviti.

U drugom pristupu, posmatra se uticaj određenog broja neslučajnih veličina, *kontrolnih faktora* na vrednost odgovarajuće slučajne veličine Y. Ova veza je deterministička (neslučajna). Na veličinu Y takođe utiču i slučajni faktori, kao i drugi neslučajni faktori čiji se uticaj ne može efektivno sagledati. Pretpostavićemo da su ova dva uticaja međusobno nezavisna, aditivna (vrednost promenljive Y je zbir vrednosti determinističke i slučajne komponente) i da je očekivanje slučajne komponente jednako nuli.

Neka su $a_1, \dots, a_p$ vrednosti kontrolnih faktora. Pretpostavićemo da ovi faktori utiču na Y posredstvom funkcija $f_i(a_1, \dots, a_p)$ gde je $i = 1, \dots, n$, pri čemu je zavisnost veličine Y od vrednosti funkcija f_i linearna. Drugim rečima pretpostavićemo da važi

$$Y = x_1 f_1(a_1, \ldots, a_p) + \ldots + x_n f_n(a_1, \ldots, a_p) + \epsilon. \tag{4.35}$$

Sa $x_1, \ldots, x_n$ označili smo vrednosti koeficijenata linearne veze, a sa ϵ slučajnu komponentu. Ovakav model se naziva model *linearne regresije druge vrste*. Potrebno je izvršiti odgovarajuću procenu koeficijenata $x_1, \ldots, x_n$ tako da se relacija (4.35) "najbolje" slaže sa dobijenim eksperimentalnim rezultatima. Pri tome se vrši niz eksperimenata sa različitim vrednostima kontrolnih faktora $a^k = (a_{k1}, \ldots, a_{kn})$ $k = 1, \ldots, m$ gde je m ukupan broj eksperimenata. Označimo sa Y_i i ϵ_i, $i = 1, \ldots, m$ dobijene vrednosti veličine Y i slučajne veličine ϵ u i-tom eksperimentu (ponavljanju). Dobija se

$$Y_i = x_1 f_1(a_{i1}, \ldots, a_{ip}) + \ldots, x_n f_n(a_{i1}, \ldots, a_{ip}) + \epsilon_i, \quad i = 1, \ldots, m. \tag{4.36}$$

Označimo sada $\mathbf{Y} = [Y_1 \ \cdots \ Y_m]^T$, $\mathbf{A} = [f_j(a_{i1}, \ldots, a_{ip})]$, $\mathbf{x} = [x_1 \ \cdots \ x_n]^T$ kao i $\epsilon = [\epsilon_1 \ \cdots \ \epsilon_m]^T$. Sistem jednačina (4.36) možemo zapisati u matričnom obliku na sledeći način

$$\mathbf{Y} = \mathbf{A}\mathbf{x} + \epsilon \tag{4.37}$$

Potrebno je, za zadate vrednosti matrice $\mathbf{A}$ i vektora $\mathbf{Y}$ naći vrednost vektora $\mathbf{x} = \mathbf{x}^*$ takvu da je norma vektora ϵ minimalna, odnosno

$$\min_{\mathbf{x} \in \mathbb{R}^{n \times 1}} \|\mathbf{Y} - \mathbf{A}\mathbf{x}\| = \|\mathbf{Y} - \mathbf{A}\mathbf{x}^*\|. \tag{4.38}$$

U zavisnosti od izbora norme $\|\cdot\|$ razlikujemo različite tipove linearne regresije drugog reda. Najčešće se uzima euklidska L_2 norma, ali se koriste i L_1, L_p kao i L_∞ norme. Kada je u pitanju L_1 i L_∞ norma, problem (4.38) svodi se na problem linearnog programiranja (videti monografiju [131] ili [30, 106]).

Pretpostavimo da je norma u (4.38) euklidska. Problem (4.38) predstavlja problem nalaženja minimalnog srednjekvadratnog rešenja sistema jednačina $A\mathbf{x} = \mathbf{Y}$. Na osnovu Teoreme 3.1.9 imamo da je rešenje $\mathbf{x}^*$ odredjeno sledećim izrazom

$$\mathbf{x}^* = A^\dagger \mathbf{Y} + (I_m - A^\dagger A)z, \quad z \in \mathbb{R}^{m \times 1} \tag{4.39}$$

Prema tome, za odredjivanje x^*, dovoljno je izračunati $A^\dagger$. Primetimo da ukoliko je matrica A potpunog ranga vrsta, odnosno ako je $\text{rank}A = m$, tada na osnovu Leme 3.1.12 važi $A^\dagger = (A^* A)^{-1} A^*$, kao i $A^\dagger A = I_m$ pa se izraz (4.39) svodi na

$$\mathbf{x}^* = (A^* A)^{-1} A^* \mathbf{Y}. \tag{4.40}$$

Ovaj izraz je dobro poznat u literaturi [30, 106].

Primer 4.7.1. Pretpostavimo da veličina Y zavisi od jednog kontrolnog faktora t i da je zavisnost oblika

$$Y = at + bt^3 + ct^5 + \epsilon. \tag{4.41}$$

Pritom, neka je dat sledeći skup vrednosti:

t_i	-1	-0.5	0.5	1
Y_i	-8.98112	-1.00373	1.04187	9.01672

Najpre formiramo matricu A i vektor $\mathbf{Y}$

$$A = \begin{bmatrix} t_1 & t_1^3 & t_1^5 \\ t_2 & t_2^3 & t_2^5 \\ t_3 & t_3^3 & t_3^5 \\ t_4 & t_4^3 & t_4^5 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -0.5 & -0.125 & -0.03125 \\ 0.5 & 0.125 & 0.03125 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} -8.98112 \\ -1.00373 \\ 1.04187 \\ 9.01672 \end{bmatrix}$$

Zatim računamo MP inverz $A^\dagger$, kao i rešenje primenom (4.39) pri čemu je $z = \mathbb{O}$,

$$A^\dagger = \begin{bmatrix} 0.0833333 & -1.14286 & 1.14286 & -0.0833333 \\ -0.25 & 0.380952 & -0.380952 & 0.25 \\ -0.333333 & 0.761905 & -0.761905 & 0.333333 \end{bmatrix}, \quad \mathbf{x}^* = A^\dagger \mathbf{Y} = \begin{bmatrix} 0.838005 \\ 3.72018 \\ 4.44073 \end{bmatrix}.$$

Prema tome, najbolja L_2 aproksimacija veličine Y oblika (4.41) je

$$Y = 0.838005t + 3.72018t^3 + 4.44073t^5$$

4.7.2 Problem projektovanja sistema sa povratnom vezom

U ovom odeljku pokazaćemo jednu primenu generalisanih inverza u teoriji automatskog upravljanja. Najpre dajemo jedan kratak uvod u teoriju linearnih sistema automatskog upravljanja. Sistem automatskog upravljanja prima ulazni signal $x(t)$ (promenljiva t ima ulogu vremena) a na njegovom izlazu se pojavljuje izlazni signal $y(t)$. Ako pretpostavimo da su i ulazni i izlazni signal neprekidne (kontinualne) funkcije vremena, tada kažemo da je sistem automatskog upravljanja *kontinualan*. Kontinualan sistem automatskog upravljanja je *kauzalan* ako iz $x(t) = 0$ za svako $t \in \mathbb{R}$ sledi da je $y(t) = 0$ za svako $t \in \mathbb{R}$. Drugim rečima, nulti ulaz u kauzalni sistem dovodi do nultog izlaza. U ovom odeljku mi ćemo pretpostaviti da su zadovoljeni uslovi kauzalnosti za sve dinamičke sisteme koje razmatramo.

Uobičajeno je da se kod kontinualnih sistema automatskog upravljanja ulazni i izlazni signali posmatraju u Laplaceovom domenu (ili s-domenu). Drugim rečima, umesto da radimo sa funkcijama $x(t)$ i $y(t)$ mi ćemo da posmatramo njihove Laplaceove transformacije

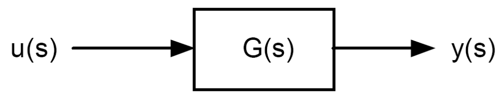
$$X(s) = \int_0^{+\infty} x(t)e^{-st}dt, \quad Y(s) = \int_0^{+\infty} y(t)e^{-st}dt. \tag{4.42}$$

Dodatne informacije o Laplaceovoj transformaciji mogu se naći u literaturi, npr. [41, 54, 56, 70]. Naravno, svaki sistem može da ima više ulaza i više izlaza pa će, u opštem slučaju, $X(s)$ i $Y(s)$ (a takođe i $x(t)$ i $y(t)$) biti vektori. Pretpostavićemo da su $X(s)$ i $Y(s)$ vektori formata $m \times 1$ i $n \times 1$. Za linearni dinamički sistem, signali $X(s)$ i $Y(s)$ su povezani linearnim relacijama, tj. tada postoji matrica $G(s)$, koja se zove *prenosna matrica*, takva da je zadovoljeno

$$Y(s) = G(s)X(s).$$

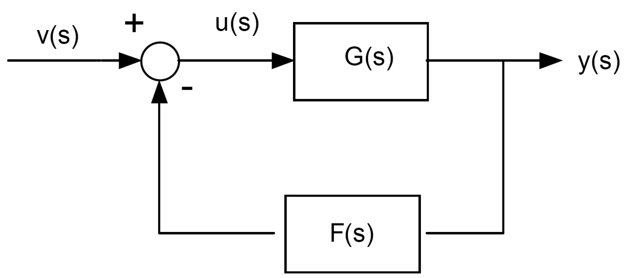
Prema tome, svaki linearni dinamički sistem je u potpunosti opisan svojom prenosnom matricom $G(s)$. Dalje, pretpostavićemo da je $G(s)$ racionalna ili polinomijalna matrica. U tom slučaju relacija $Y(s) = G(s)X(s)$ u vremenskom domenu predstavlja sistem integro-diferencijalnih jednačina.

Linearni dinamički sistemi se obično predstavljaju blok dijagramima. Na Slici 4.1 prikazan je blok dijagram najjednostavnijeg linearnog sistema. Ceo sistem je predstavljen jednim blokom sa prenosnom matricom $G(s)$ i ulaznim i izlaznim signalima $U(s)$ i $Y(s)$. Više blokova mogu da se kombinuju da bi se konstruisao složeniji linearni sistem.



Slika 4.1: Blok dijagram najjednostavnijeg linearnog sistema

Na slici 4.2 predstavljen je poznat sistem sa povratnom vezom. Izlazni signal $Y(s)$ se vraća nazad na ulaz u sistem (povratna veza) i dodaje se ulaznom signalu $V(s)$ tako da se prenosna matrica $F(s)$ primenjuje i na ulazni signal $V(s)$ i na povratni signal $Y(s)$. Ovaj koncept prvi je uveo Harold Stephen Black 1927. godine za stabilizaciju električnih pojačavača. Princip povratne veze se široko koristi u mnogim praktičnim primenama u elektrotehnici i mašinstvu.



Slika 4.2: Blok dijagram sistema sa povratnom vezom

Više informacija o povratnoj vezi, a takođe i o linearnim dinamičkim sistemima može se naći u literaturi, npr. [73, 74, 95, 139].

Posmatrajmo linearni sistem prikazan na slici 4.1 gde je $G(s)$ racionalna ili polinomijalna prenosna matrica formata $m \times n$. Zadatak je da se odredi (ukoliko postoji) matrica $F(s)$ tako da sistem sa povratnom vezom prikazan na slici 4.2 ima željenu prenosnu matricu $H(s) \in \mathbb{C}^{m\times n}(s)$.

Tada važi $H(s) = (I_m + G(s)F(s))^{-1}G(s)$ i sledeća jednakost mora da bude zadovoljena

$$G(s)F(s)H(s) = G(s) - H(s), \tag{4.43}$$

gde je $F(s)$ nepoznata matrica. Naredna teorema daje potrebne i dovoljne uslove za postojanje rešenja problema kompenzacije povratne veze za date matrice $G(s)$ i $H(s)$.

Teorema 4.7.1. *Potreban i dovoljan uslov da matrična jednačina (4.43) ima rešenje dat je sa*

$$G(s) = G(s)H(s)^{\dagger}H(s), \quad H(s) = G(s)G(s)^{\dagger}H(s). \tag{4.44}$$

Dokaz. Primenom teoreme 3.1.17 dobijamo sledeći potreban i dovoljan uslov za postojanje rešenja jednačine (4.43)

$$G(s)G(s)^{\dagger}(G(s)-H(s))H(s)^{\dagger}H(s)=G(s)-H(s). \tag{4.45}$$

Poslednja relacija može dalje da se uprosti tako da dobije sledeći oblik

$$G(s)(I_n-H(s)^{\dagger}H(s))=(I_m-G(s)G(s)^{\dagger})H(s). \tag{4.46}$$

Množenjem (4.46) sa $H(s)^{\dagger}$ sa desne strane, dobijamo

$$(I_m-G(s)G(s)^{\dagger})H(s)H(s)^{\dagger}=G(s)(H(s)^{\dagger}-H(s)^{\dagger}H(s)H(s)^{\dagger})=\mathbb{O},$$

odnosno važi

$$H(s)H(s)^{\dagger}=G(s)G(s)^{\dagger}H(s)H(s)^{\dagger}. \tag{4.47}$$

Ponovnim množenjem (4.47) sa $H(s)$ sa desne strane dobijamo

$$H(s)=G(s)G(s)^{\dagger}H(s). \tag{4.48}$$

Analogno, može da se dokaže da je

$$G(s)=G(s)H(s)^{\dagger}H(s). \tag{4.49}$$

Napomenimo da iz relacija (4.48) i (4.49) sledi relacija (4.46). Štaviše, dobijamo da su obe strane relacije (4.48) jednake $\mathbb{O}$. Ovo dokazuje da je relacija (4.46) ekvivalentna sa sistemom jednačina (4.48) i (4.49). $\square$

Prema tome, da bi utvrdili da li postoji odgovarajuća matrica povratne sprege, moramo proveriti da li važi uslov (4.44), odnosno moramo izračunati MP inverze $G(s)^{\dagger}$ i $H(s)^{\dagger}$. Za ovo izračunavanje možemo da koristimo neki od algoritama iz prethodnih sekcija. Napomenimo da ako je $A(s)\in\mathbb{C}^{m\times n}(s)$ racionalna matrica, ona može da se predstavi kao

$$A(s)=\frac{1}{a_1(s)}A_1(s) \tag{4.50}$$

gde je $A_1(s)\in\mathbb{C}^{m\times n}[s]$ polinomijalna matrica a polinom $a_1(s)$ je jednak najmanjem zajedničkom sadržaocu svih imenioca u $A(s)$. Sada MP inverz $A(s)^{\dagger}$ može da se predstavi kao $A(s)^{\dagger}=a_1(s)A_1(s)^{\dagger}$. Koristeći relaciju (4.50) izračunavanje generalisanog inverza racionalne matrice $A(s)$ možemo da svedemo na izračunavanje generalisanog inverza za polinomijalnu matricu $A_1(s)$. Isto važi i za Drazinov inverz i neke druge klase generalisanih inverza. Ova činjenica omogućava da se iskoriste metodi razvijeni za polinomijalne matrice (na primer Algoritam 4.2.1, Algoritam 4.3.1, Algoritam 4.4.1, itd.) za izračunavanje generalisanih inverza racionalnih matrica.

Ako je relacija (4.44) zadovoljena, tj. ako postoji rešenje problema povratne veze, saglasno Teoremi 3.1.17, sva rešenja jednačine (4.43) opisana su relacijom

$$F(s) = G(s)^{\dagger}(G(s) - H(s))H(s)^{\dagger} + Y(s) - G(s)^{\dagger}G(s)Y(s)H(s)H(s)^{\dagger}. \tag{4.51}$$

Od svih matrica $F(s)$ definisanih sa (4.51) izdvojićemo samo one za koje je matrica $I_m + G(s)F(s)$ regularna (u suprotnom, sistem sa povratnom vezom prikazan na slici 4.2 osciluje i izlaz $y(t)$ je nezavisan od ulaza $x(t)$).

Primer 4.7.2. [64]

Razmotrimo sledeću prenosnu matricu nekog sistema

$$G(s) = \begin{bmatrix} \frac{1}{s-2} & 0 & 0 \\ 0 & \frac{1}{s-1} & 0 \end{bmatrix}.$$

Potrebno je da utvrdimo da li postoji, i ako postoji da odredimo matricu povratne veze $F(s)$ tako da sistem sa povratnom vezom ima sledeću prenosnu matricu

$$H(s) = \begin{bmatrix} 0 & \frac{1}{s+1} & 0 \\ \frac{1}{(s+1)^2} & 0 & 0 \end{bmatrix}.$$

Prvo ćemo da izračunamo generalisane inverze $G(s)^{\dagger}$ i $H(s)^{\dagger}$. Te vrednosti su

$$G(s) = \begin{bmatrix} s-2 & 0 \\ 0 & s-1 \\ 0 & 0 \end{bmatrix}, \qquad H(s) = \begin{bmatrix} 0 & (s+1)^2 \\ s+1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Direktna provera daje da su relacije $H(s) = G(s)G(s)^{\dagger}H(s)$ i $G(s) = G(s)H(s)^{\dagger}H(s)$ zadovoljene. Saglasno Teoremi 4.7.1, postoji rešenje našeg problema povratne veze. Iskoristićemo izraz (4.51) da bi odredili vrednost $F(s)$. Neka je $Y(s) = \mathbb{O}$. Dobijeno rešenje je

$$F(s) = G(s)^{\dagger}(G(s) - H(s))H(s)^{\dagger} = \begin{bmatrix} 2-s & (s+1)^2 \\ s+1 & 1-s \\ 0 & 0 \end{bmatrix}.$$

Takođe, direktnom proverom zaključujemo da je matrica $I_2 + G(s)F(s)$ regularna, što znači da je $F(s)$ stvarno rešenje našeg problema povratne veze.

Na kraju ovog odeljka napomenimo da sličan postupak može da se primeni na znatno složenije sisteme automatskog upravljanja.

Glava 5

Zaključak

U ovoj doktorskoj disertaciji analizirane su dve primene metoda simboličkog izračunavanja:

(1) Simboličko izračunavanje Hankelovih determinanti (tj Hankelove transformacije) i

(2) Simboličko izračunavanje generalisanih inverza konstantnih, racionalnih i polinomijalnih matrica.

Oblasti u kojima su ostvarene ove primene se medjusobno veoma razlikuju. Medjutim, uspešna primena metoda simboličkog računanja je ono što je zajedničko obema disciplinama. Glavni cilj ove doktorske disertacije je bio upravo detaljna analiza mogućnosti i načina primene simboličkog izračunavanja za uspešno računanje kako Hankelovih transformacija mnogih nizova, tako i široke klase generalisanih inverza konstantnih, racionalnih i polinomijalnih matrica. Činjenica je da je ova oblast popriličnо neistražena i da još uvek ima dosta prostora za dalja istraživanja.

Kratak pregled svih izloženih rezultata kao i neke ideje za dalja istraživanja biće prikazani u ovoj, zaključnoj glavi.

A. Proučeni su metodi za simboličko izračunavanje Hankelovih transformacija nizova i detaljno su opisana izračunavanja Hankelove transformacije nekoliko klasa nizova celih i realnih brojeva. Celine koje smo pritom obradili, kao i rezultati i zaključci do kojih smo došli, mogu se sistematizovati na sledeći način:

[**A.1.**] Najpre su date osnovne definicije i pojmovi vezani za nizove realnih brojeva, sa posebnim osvrtom na nizove celih brojeva. Takodje detaljno su opisane tri najvažnije trasformacije brojevnih nizova uključujući i Hankelovu transformaciju. Prikazana su osnovna svojstva ovih transformacija kao i jedna primena Hankelove transformacije u fizici čvrstog stanja.

[**A.2.**] Formulisani su osnovni metodi za izračunavanje Hankelove transformacije. To su metod Dodgsonove kondenzacije i Radoux-Junodov metod.

[**A.3.**] Da bi formulisali metod za računanje Hankelove transformacije baziran na verižnim razlomcima i ortogonalnim polinomima, bilo je potrebno izvršiti pregled najvažnijih definicija i tvrdjenja iz teorije ortogonalnih polinoma. Poseban osvrt bio je na tročlanoj rekurentnoj relaciji kao i svojstvima koja proističu iz nje.

[**A.4.**] Nakon toga je detaljno opisan metod baziran na ortogonalnim polinomima i verižnim razlomcima. Sve etape ovog metoda su detaljno formulisane. Kompletan metod je na kraju formulisan u obliku algoritma. Metode za nalaženje težinske funkcije primenom Stieltjesove inverzione formule kao i metode za trasformaciju težinske funkcije su, takodje, vrlo detaljno opisane.

B. Izloženi metodi za računanje Hankelove transformacije primenjeni su na konkretnim klasama nizova kao i nekim tvrdjenjima vezanim za odnos Hankelove i drugih nizovnih transformacija. Napomenimo da su prilikom svih izvodjenja intenzivno korišćene metode simboličkog računanja kao i programski paket MATHEMATICA. Svi rezultati koji pripadaju ovoj grupi su **originalni** (pri čemu su neki od njih već publikovani u medjunarodnim časopisima) i mogu se sistematizovati na sledeći način:

[**B.1.**] Metod baziran na ortogonalnim polinomima primenjen je na izračunavanje Hankelove transformacije niza čiji je opšti član jednak sumi dva uzastopna generalisana Catalanova broja. Najpre je odredjena težinska funkcija primenom Stieltjesove inverzione formule a zatim su primenom niza trasformacija ove težinske funkcije odredjeni koeficijenti tročlane rekurentne relacije. Na kraju je dokazan glavni rezultat kojim je, u zatvorenom obliku, odredjena Hankelova transformacija posmatranog niza. Ovi rezultati su publikovani u našem radu [113]. Ovo je jedan od glavnih rezultata ove disertacije.

[**B.2.**] Proučen je odnos izmedju Hankelove i k-binomnih transformacija. Ove transformacije predstavljaju uopštenje binomne transformacije. Glavni rezultat ovog dela je invarijantnost Hankelove u odnosu na opadajuću binomnu transformaciju kao i jednostavna formula koja povezuje Hankelove transformacije originalnog niza i niza trasformisanog rastućom k-binomnom transformacijom. Ovi rezultati se ovom prilikom prvi put objavljuju.

[**B.3.**] Izračunata je i Hankelova transformacija niza generalisanih centralnih trinomnih koeficijenata. Pritom je korišćen modifikovani metod baziran na ortogonalnim polinomima kao i predhodno dobijeni rezultati vezani za odnos Hankelove i k-binomnih transformacija. Ovi rezultati se ovom prilikom prvi put objavljuju.

[**B.4.**] Radoux-Junodov metod primenjen je na računanje Hankelove transformacije inverzije niza generalisanih Fibonaccijevih brojeva. Dato je i nekoliko generalizacija dobijenih rezultata.

C. Detaljno su proučeni metodi za simboličko računanje generalisanih inverza konstantnih matrica. Takodje, izložene su osnove teorije genealisanih inverza matrica. Svi metodi su implementirani u simboličkom programskom jeziku MATHEMATICA. Celine koje smo pritom obradili, kao i rezultati i zaključci do kojih smo došli, mogu se sistematizovati na sledeći način:

[**C.1.**] Dat je pregled osnovnih rezultata iz linearne algebre i teorije matrica, koji su korišćeni u daljem radu sa posebnim osvrtom na nekoliko klasičnih dekompozicija matrica. Definisano je nekoliko klasa uopštenih inverza i proučena su njihova osnovna svojstva. Posebna pažnja posvećena je Moore-Penroseovom, težinskom Moore-Penroseovom i Drazinovom inverzu.

[**C.2.**] Prezentovani su i detaljno proučeni osnovni metodi za računanje generalisanih inverza. To su metodi bazirani na faktorizacijama potpunog ranga, metodi bazirani na blokovskim reprezentacijama i metod Žukovskog.

[**C.3.**] Prikazan je i detaljno proučen Leverrier-Faddev metod, odnosno modifikacije ovog metoda za računanje Moore-Penroseovog, Drazinovog i široke klase ostalih generalisanih inverza. Formulisano je nekoliko varijanti ovog metoda koje su predstavljene u obliku algoritama. Za svaki algoritam diskutovana je vremenska složenost.

[**C.4.**] Metodi pregradjivanja su bili predmet daljih proučavanja. Formulisane su tri varijante ovog metoda za računanje Moore-Penroseovog inverza, $\{1\}$ inverza i težinskog Moore-Penroseovog inverza. I ove varijante su date u obliku algoritma kojima je zatim odredjena vremenska složenost.

D. Jedan od glavnih rezultata ove disertacije je metod za računanje Moore-Penroseovog i $\{i, j, \dots, k\}$ inverza u vremenu množenja matrica. Ova složenost je ujedno i teorijski najbolja složenost koju može imati algoritam za računanje generalisanih inverza. Metod je baziran na modifikaciji Courrierovog metoda i generalisanoj Cholesky faktorizaciji. Ovi rezultati su originalni i još uvek neobjavljeni.

E. Dalje su proučavani metodi za simboličko računanje generalisanih inverza racionalnih i polinomijalnih matrica. To su modifikacije Leverrier-Faddevog metoda i metoda pregradjivanja. Dobijeni rezultati su originalni i većinom publikovani u našim radovima [100, 101, 102, 103, 104, 130, 141].

[**E.1.**] Dat je pregled najvažnijih definicija i pomoćnih lema vezanih za racionalne i polinomijalne matrice.

[**E.2.**] Najpre je prikazan interpolacioni metod za računanje Moore-Penroseovog inverza polinomijalnih matrica. Ovaj metod je baziran na Leverrier-Faddevom metodu. Izračunate su vremenske složenosti Leverrier-Faddevog metoda primenjenog na polinomijalne matrice kao i interpolacionog metoda. Takodje je dat jedan jednostavan metod za procenu stepena odgovarajućih polinomijalnih matrica. Implementacije ovih algoritama u programskom paketu MATHEMATICA testirane su na slučajno generisanim test primerima i rezultati testiranja su prokomentarisani. Tako je i u praksi potvrdjen teorijski dobijen rezultat da je interpolacioni metod uspešniji od klasičnog, naročito kada je ulazna matrica gusta. Ovi rezultati su originalni i preuzeti iz našeg rada [130].

[**E.3.**] Slična ideja je iskorišćena za konstrukciju inteprolacionog metoda za računanje Drazinovog inverza polinomijalnih matrica baziranog na Leverrier-Faddevom metodu. Konstruisan je interpolacioni metod za računanje Drazinovog inverza i izračunata je njegova vremenska složenost. Implementacija interpolacionog metoda u programskom paketu MATHEMATICA testirana je na slučajno generisanim test primerima i rezultati testiranja su prokomentarisani. Tako je i u praksi potvrdjen teorijski dobijen rezultat da je interpolacioni metod uspešniji od klasičnog naročito kada je ulazna matrica gusta. Ovi rezultati su originalni i preuzeti iz našeg rada [103].

[**E.4.**] Nakon toga je konstruisan interpolacioni metod za računanje široke klase generalisanih inverza. Takođe, konstruisani su interpolacioni metodi za računanje indeksa i ranga polinomijalne matrice. Ovi rezultati su originalni i preuzeti iz našeg rada [102].

[**E.5.**] Razmatrana je modifikacija metoda pregrađivanja za računanje Moore-Penroseovog inverza racionalnih i polinomijalnih matrica sa dve promenljive. Modifikacija je implementirana u programskom paketu MATHEMATICA. Ovi rezultati su originalni i preuzeti iz naših radova [100, 101].

[**E.6.**] Prikazana je modifikacija metoda pregrađivanja za računanje težinskog Moore-Penroseovog inverza racionalnih i polinomijalnih matrica. Definisane su i retke strukture kojima se izloženi metodi značajno ubrzavaju. Ovi rezultati su originalni i preuzeti iz naših radova [141, 104].

F. Razmatrane su primene teorije generalisanih inverza konstantnih i polinomijalnih matrica u matematičkoj statistici i automatici (odnosno teoriji automatskog upravljanja). Deo vezan za primene u automatici sadrži nekoliko originalnih, još uvek neobjavljenih rezultata.

Na samom kraju ove doktorske disertacije, napomenimo da bi dalja istraživanja na temu simboličkog izračunavanja Hankelovih determinanti i generalisanih inverza matrica mogla da se odvijaju u sledećim pravcima:

(1) Konstrukcija metoda (tj. implementacija) kojim se potpuno automatizuje primena metoda baziranog na ortogonalnim polinomima za računanje Hankelove transformacije. Neki delovi ovog softvera su već napisani.

(2) Uspostavljanje veze između nekih drugih nizovnih transformacija i Hankelove transformacije, kao što je to bio slučaj sa k-binomnim transformacijama.

(3) Izračunavanje Hankelovih transformacija još nekih nizova (niz Narayaninih polinoma, zatvorenih šetnji regularnih stabala, itd...).

(4) Konstrukcija metoda za računanje Drazinovog i još nekih generalisanih inverza u vremenu množenja matrica.

(5) Konstrukcija interpolacionih metoda za računanje generalisanih inverza racionalnih matrica.

U tom smislu, rezultati izloženi u ovom radu, pored značajnog doprinosa simboličkom izračunavanju Hankelovih determinanti i generalisanih inverza matrica, predstavljaju i dobru osnovu za dalja istraživanja i razvoj novih algoritama za simboličko izračunavanje u drugim oblastima.

Literatura

[1] N.I. Akhiezer, *The classical moment problem and some related questions in analysis*, Translated by N. Kemmer, Hafner Publishing Co., New York, 1965.

[2] A.S. Cvetković, G.V. Milovanović, *The MATHEMATICA package "OrthogonalPolynomials"* Facta Univerzitatis (Niš), Ser. Math. Inform. **19** (2004), 17–36

[3] S.M. Balle, P.C. Hansen, N.J. Higham, *A Strassen-type matrix inversion algorithm*, Danish Computing Center for Research and Education, Building 305, Technical University of Denmark, DK-2800 Lyngby, Denmark, PaA2 Deliverable APPARC ESPRIT Contract, 1993.

[4] T. Banachiewicz, *Zur Berechnung der Determinanten, wie auch der Inversen und zur darauf basierten Auflosung der Systeme linearer Gleichungen*, Acta Astronom. Ser. C 3 (1937), 41-67.

[5] S. Barnett, *Leverrier's algorithm: a new proof and extensions* SIAM J. Matrix Anal. Appl. **10** (1989), 551–556.

[6] P. Barry, *On Integer Sequences Based Constructions of Generalized Pascal Triangles*, Preprint, Waterford Institute of Technology, 2005.

[7] P. Barry, *A conjecture on the form of the Hankel transform of the sum of consecutive generalized Catalan numbers*, unpublished manuscript.

[8] P. Barry, P.M. Rajković, M.D. Petković, *On the Hankel transform of generalized central trinomial coefficients*, unpublished manuscript.

[9] P. Barry, P.M. Rajković, M.D. Petković, *Hankel Transform of the Reversion of Generalized Fibonacci Sequences*, unpublished manuscript.

[10] A. Ben-Israel and T.N. E. Greville, *Generalized Inverses. Theory and Applications, Second edition*, CMS Books in Mathematics/Ouvrages de Mathmatiques de la SMC, 15. Springer-Verlag, New York, 2003.

[11] C. Berg, *Indeterminate moment problems and the theory of entire functions*, J. Comput. Appl. Math. **65** (1995), no. 1, 27-55.

[12] R. Brualdi, S. Kirkland, *Aztec diamonds and digraphs, and Hankel determinants of Schröder numbers*. Journal of Combinatorial Theory, Series B **94** (2005) 334 - 351

[13] F. Bu, Y. Wei, *The algorithm for computing the Drazin inverses of two-variable polynomial matrices*, Appl. Math. Comput. **147** (2004) 805–836.

[14] F. Burns, D. Carlson, E. Haynsworth, T. Markham, *Generalized inverse formulas using Schur complement*, SIAM J. Appl. Math, **26**, No. 2, March 1974.

[15] S. L. Campbell (Editor), *Recent Applications of Generalized Inverses*, Pitman, London, 1982.

[16] S. L. Campbell, C. D. Meyer, *Generalized inverses of Linear Transformations*, Pitman, New York, 1979.

[17] T.S. Chihara, *An Introduction to Orthogonal Polynomials*, Gordon and Breach, New York, 1978.

[18] T.S. Chihara, , *Hamburger moment problems and orthogonal polynomials*, Trans. Amer. Math. Soc. Vol **315** (1989), no. 1, 189-203.

[19] R.E. Cline, *Representation of the generalized inverse of a partitioned matrix*, J. Soc. Indust. Appl. Math., **12** (1964), pp.588-600.

[20] F. Colomo, A.G. Pronko, *Square ice, alternating sign matrices, and classical orthogonal polynomials*, J. Stat. Mech. (2005) P01005.

[21] E.F. Conrad, *Some continued fraction expansions of Laplace transforms of elliptic functions*, phd. thesis, Ohio State University, Ohio, 2002.

[22] D. Coppersmith, S. Winograd, *Matrix multiplication via arithmetic progression*, J. Symbolic Comput., **9** (1990), 251–280.

[23] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms, Second Edition*, The MIT Press, Cambridge, Massachusetts London, McGraw-Hill Book Company, Boston Burr Ridge , IL Dubuque , IA Madison , WI New York San Francisco St. Louis Montreal Toronto, 2001.

[24] P. Courrieu, *Fast Computation of Moore-Penrose Inverse Matrices* Neural Information Processing - Letters and Reviews, Vol **8** No 2 (2005) 25–29.

[25] P. Courrieu *Straight monotonic embedding of data sets in Euclidean spaces* Neural Network, **15** (2002) 1185–1196.

[26] A. Cvetković, P. Rajković, M. Ivković, *Catalan Numbers, the Hankel Transform and Fibonacci Numbers*, Journal of Integer Sequences, **5**, May 2002, Article 02.1.3.

[27] H. P. Decell, *An application of the Cayley-Hamilton theorem to generalized matrix inversion*, SIAM Review **7** No 4 (1965) 526–528.

[28] M.P. Drazin, *Pseudo inverses in associative rings and semigroups*, Amer. Math. Monthly **65**, (1958), 506514.

[29] C.L. Dodgson, *Condensation of determinants*, Proc. Royal Soc. London **15** (1866), 150–155.

[30] A.L. Edwards, *Multiple Regression and the Analysis of Variance and Covariance*, San Francisco, CA, W.H. Freeman, 1979.

[31] O. Eğecioğlu, T. Redmond, C. Ryavec, *Almost product evaluation of Hankel determinants* The Electronic Journal of Combinatorics **15** (2008), #R6

[32] I. Erdelyi, *On the Matrix Equation* $Ax = \lambda Bx$ J. Math. Anal. Appl. **17**, No 1 (1967), 119–132.

[33] L. Euler, *Observationes analyticae*, Novi Commentarii Acad. Sci. Imper. Petropolitanae, **11** (1765) 1767, 124–143.
Reprinted in *Opera Omnia*, Series I, vol. 15, 50–69.

[34] D.K. Faddeev, V.N. Faddeeva, Computational methods of linear algebra (in Russian) Fizmatgiz, Moscow, 1960. (English translation by R. C. Williams, W.H. Freeman, San Francisco, 1963.)

[35] G. Fragulis, B.G. Mertzios, A.I.G. Vardulakis, *Computation of the inverse of a polynomial matrix and evaluation of its Laurent expansion* Int. J. Control **53** (1991) 431–443.

[36] J.S. Frame, *A simple recursion formula for inverting a matrix*, Bull. Amer. Math. Soc. **55** (1949), 19-45.

[37] C. French, *Transformations Preserving the Hankel Transform*, Journal of Integer Sequences, Vol **10** (2007), Article 07.7.3.

[38] R. Gabriel, *Das verallgemeinerte inverse einer matrix, deren elemente einem Körper angehören* J. Rewie Ansew Math., **234** (1969), 107–122.

[39] W. Gautschi, *Orthogonal polynomials: applications and computations*, in *Acta Numerica, 1996*, Cambridge University Press, 1996, pp. 45–119.

[40] W. Gautschi, *Orthogonal Polynomials: Computation and Approximation*, Clarendon Press - Oxford, 2003.

[41] U. Graf, *Applied Laplace Transforms and z-Transforms for Scientists and Engineers: A Computational Approach using a Mathematica Package*, Basel, Switzerland, Birkhauser, 2004.

[42] R.L. Graham, D.E. Knuth, O. Patashnik, *Concrete Mathematics*, Addison-Wesley, Reading, MA, 1994, 2nd edition.

[43] T.N.E. Greville, *The Souriau-Frame algorithm and the Drazin pseudoinverse*, Linear Algebra Appl. **6** (1973) 205–208.

[44] T.N.E. Greville, *Some applications of the pseudoinverse of a matrix*, SIAM Rev. 2 (1960), 1522.

[45] C. W. Groetsch, *Representation of the generalized inverse*, Journal Math. Anal. Appl. **49** (1975), 154-157.

[46] C. W. Groetsch, *Generalized inverses of linear operators*, Marcel Dekker, Inc. New York and Basel, 1977.

[47] J.C Gower, *A modified Leverrier-Faddeev algorithm for matrices with multiple eigenvalues*, Linear Algebra and its Applications, **31**, 1980, 61–70.

[48] J.C. Gower, *An application of the Leverrier-Faddeev algorithm to skew-symmetric matrix decomposition*, Utilitas Mathematica, **38**, 1980, 225-240.

[49] F.G. Gustavson, *Recursion leads to automatic variable blocking for dense linear algebra algorithms*, IBM J. Res. Develop., **41(6)**, 737–755, 1997.

[50] F.G. Gustavson, I. Jonsson, *Minimal-storage high-performance Cholesky factorization via recursion and blocking*, IBM J. Res. Develop., **44(6)**, 823–850, 2000.

[51] R. E. Hartwig, *More on the Souriau-Frame algorithm and the Drazin inverse*, SIAM J. Appl. Math. **31** No 1 (1976) 42–46.

[52] R.E. Hartwig, *A method for calculating* A^d, Math. Japonica **26**, No 1 (1981), 37–43.

[53] H.A. Helfgott, I.M. Gessel, *Enumeration of Tilings of Diamonds and Hexagons with Defects*, The Electronic Journal of Combinatorics, **6** (1999), #R16.

[54] P. Henrici, *Applied and Computational Complex Analysis, Vol. 2: Special Functions, Integral Transforms, Asymptotics, Continued Fractions*, New York, Wiley, 1991.

[55] N.J. Highman, *Exploiting fast matrix multiplication within the level 3 BLAS*, ACM Trans. Math. Software, **16** (1990), 352–368.

[56] J.C. Jaeger, G.H. Newstead, *An Introduction to the Laplace Transformation with Engineering Applications*, London, Methuen, 1949.

[57] J. Ji, *A finite algorithm for the Drazin inverse of a polynomial matrix*, Appl. Math. Comput. **130** (2002) 243–251.

[58] J. Ji, *An alternative limit expression of Drazin inverse and its application*, Appl. Math. Comput. **61** (1994), 151–156.

[59] J. Ji, *Explicit expressions of the generalized inverses and condensed Cramer rules*, Linear Algebra Appl., **404** (2005), 183–192.

[60] J. Jones, N. P. Karampetakis, A. C. Pugh, *The computation and application of the generalized inverse vai Maple*, J. Symbolic Computation,**25** (1998) 99–124.

[61] A. Junod, *Hankel determinants and orthogonal polynomials*, Expo. Math. **21** (2003), 63–74.

[62] K. Kajiwara, T. Masuda, M. Noumi, Y. Ohta, Y. Yamada Y *Determinant formulas for the Toda and discrete Toda equations*, Funkcial. Ekvac. Vol **44** (2001), 291-307.

[63] K. Kajiwara, M.Mazzocco, Y. Ohta, *A Remark on the Hankel Determinant Formula for Solutions of the Toda Equation*, Journal of Physics A: Mathematical and Theoretical, Vol **40** (2007), Issue 42, 12661-12675.

[64] N.P. Karampetakis, *Computation of the generalized inverse of a polynomial matrix and applications* Linear Algebra Appl. **252** (1997) 35–60.

[65] N.P. Karampetakis, *Generalized inverses of two-variable polynomial matrices and applications* Circuits Systems Signal Processing **16** (1997) 439–453.

[66] N.P. Karampentakis, *Computation of generalized inverse of polynomial matrix and applications*, Linear Algebra Appl. **252**(1997), 35–60.

[67] N.P. Karampetakis, P. Tzekis, *On the computation of the generalized inverse of a polynomial matrix*, Ima Journal of Mathematical Control and Information **18** (2001) 83–97.

[68] N.P. Karampentakis, S. Vologianidis, *DFT calculation of generalized and Drazin inverse of polynomial matrix*, Appl. Math. Comput. **143**(2003), 501–521.

[69] N.P. Karampetakis, *Generalized inverses of two-variable polynomial matrices and applications*, Circuits Systems Signal Processing **16** (1997) 439–453.

[70] S.G. Krantz, *The Laplace Transform. 15.3 in Handbook of Complex Variables*, Boston, MA, Birkhauser, pp. 212–214, 1999.

[71] C. Krattenthaler, *Advanced determinant calculus*, Seminaire Lotharingien Combin. **42** ("The Andrews Festschrift") (1999), Article B42q, 67 pp.

[72] C. Krattenthaler, *Advanced determinant calculus: a complement*, Linear Algebra Appl. **411** (2005), 68–166.

[73] V. Kučera, *Diophantine Equations in Control - a survey*, Automatica, **29** (1993), 1361–1375.

[74] B.C. Kuo, F. Golnaraghi, *Automatic control systems, Eighth edition*, NY, Wiley, ISBN 0471134767, 2003.

[75] N. Krejić, Dj. Herceg, *Matematika i MATHEMATICA*, Institut za matematiku, Novi Sad, 1993.

[76] Wolfgang Lang, *On sums of powers of zeros of polynomials*, Journal of Comput. Appl. Mathematics, Vol. **89**, Issue 2 (March 1998), 237 – 256.

[77] J.W. Layman, *The Hankel Transform and Some of its Properties,* Journal of Integer Sequences, Article 01.1.5, Volume 4, 2001.

[78] U.J.J. Leverrier, *Sur les variations seculaires des elements des orbites*, J. Math., 1840.

[79] X. Li, Y. Wei, *A note on computing the generalized inverse $A^{(2)}_{T,S}$ of a matrix A*, Int. J. Math. Math. Sci., **31** (2002), 497–507.

[80] V. Lovass Nagy, R. Miller, D. Powers, *Transfer function matrix synthesis by matrix generalized inverses* Int. J. Control **27** (1978) 387–391.

[81] V. Lovass Nagy, R. Miller, D. Powers, *Further results on output control in the servomechanism sence* Int. J. Control **27** (1978) 133–138.

[82] V. Lovass Nagy, R. Miller, D. Powers, *An introduction to the application of the simplest matrix-generalized inverse in system science* IEEE Trans. Auto. Control **25** (1978) 766–771.

[83] H. Lütkepohl, *Handbook of Matrices*, New York: Wiley, 1996.

[84] C.C. MacDuffee, *The Theory of Matrices*, Chelsea, New York, N.Y., 1956.

[85] AR. Meenakshi, N. Anandam, *Polynomial generalized inverses of a partitioned polynomial matrix*, Journal of the Indian Math. Soc., Vol 58, No. 1, pp. 11-18, 1992.

[86] J. Miao, *Some results for computing the Drazin inverse of a partitioned matrix*, J. Shanghai Normal Univ., **18** (1989), pp. 25–31.(Chinese)

[87] J. Miao, *Representations for the weighted Moore-Penrose inverse of a partitioned matrix*, J. Comput. Math., **7**(1989), pp.320-323.

[88] J. Miao, *General expressions for the Moore-Penrose inverse of a 2 2 block matrix*, Linear Algebra Appl., **151** (1991), pp. 1–15.

[89] T. Miwa, M. Jimbo, E. Date, *Solitons: Differential equations, symmetries and infinite dimensional algebras*, Cambridge tracts in mathematics, Vol. **135**, Cambridge University Press, Cambridge.

[90] E. Moore, *On the reciprocal of the general algebraic matrix*, Bull. Amer. Math. Soc. **26** (1920), 394-395.

[91] M.Z. Nashed, ed. *Generalized Inverses and Applications*, Academic Press, New York, 1973.

[92] M.Z. Nashed and X. Chen, *Convergence of Newton-like methods for singular operator equations using outer inverse*, Numer. Math. **66** (1993), 235–257.

[93] P. Noble, *A methods for computing the generalized inverse of a matrix*, SIAM J. Numer. Anal. **3** (1966) 582–584.

[94] T. D. Noe, *On the Divisibility of Generalized Central Trinomial Coefficients*, Journal of Integer Sequences, Vol. 9 (2006), Article 06.2.7.

[95] G. Palumbo, S. Pennisi, *Feedback amplifiers: theory and design*, Boston, Dordrecht, London, Kluwer Academic, ISBN 0792376439, 2002.

[96] P. Peart and W. J. Woan, *Generating functions via Hankel and Stieltjes matrices,* Journal of Integer Sequences, Article 00.2.1, Issue 2, Volume 3, 2000.

[97] R. Penrose, *A generalized inverse for matrices*, Proc. Cambridge Philos. Soc. **51** (1955), 406-413.

[98] M.D. Petković, P.M. Rajković, *The Hankel transform of shifted Narayana polynomials*, Proceedings of the conference PRIM, Novi Sad, 2007.

[99] M.D. Petković, P.M. Rajković, *Alternative proof of the theorems connecting Hankel transform and k-binomial transforms*, unpublished manuscript.

[100] M.D. Petković, P.S. Stanimirović, *Partitioning method for two-variable rational and polynomial matrices*, Mathematica Balcanica vol. 19, 2005, pp. 185–194.

[101] M.D. Petković, P.S. Stanimirović, *Symbolic computation of the Moore-Penrose inverse using partitioning method*, International Journal of Computer Mathematics, 82(March 2005), pp. 355–367.

[102] M.D. Petković, P.S. Stanimirović, *Interpolation algorithm of Leverrier-Faddev type for polynomial matrices*, Numerical Algorithms, **42** (2006), 345–361.

[103] M.D. Petković, P.S. Stanimirović, *Interpolation algorithm for computing Drazin inverse of polynomial matrices*, Linear Algebra and its applications, **422** (2007), 526–539.

[104] M.D. Petković, P.S. Stanimirović, *Effective partitioning method for computing weighted Moore-Penrose inverse*, Computers & Mathematics with Applications, **55** (2008), Issue 8, 1720–1734.

[105] M.D. Petković, P.S. Stanimirović, *Generalized inversion is not harder than matrix multiplication*, unpublished manuscript.

[106] B. Popović, *Matematička statistika i statističko modelovanje*, PMF Niš, 2003.

[107] W.H. Press, S.A. Teukolsky, W.T. Wetterling, B.P. Flannery, *Numerical receipts in C*, Cambridge University Press, Cambridge (MA), 1992.

[108] C.R. Rao, S.K. Mitra, *Generalized Inverse of Matrices and its Applications*, John Wiley and Sons, Inc, New York, London, Sydney, Toronto, 1971.

[109] M. Radić, *Some contributions to the inversions of rectangular matrices*, Glasnik Matematički, **1 (21)** (1966) 23–37.

[110] Ch. Radoux, *Addition formulas for polynomials built on classical combinatorial sequences*, J. of Computational and Applied Mathematics, 115 (2000) pp. 471–477

[111] Ch. Radoux, *Determinant de Hankel construit sur les polynomes de Hermite*, Annales de la Societe Scientifiques de Bruxelles, 104 (2) (1991), pp. 59–61

[112] Ch. Radoux, *The Hankel Determinant of Exponential Polynomials:A very short proof and a new result concerning Euler numbers*, Amer. Math. Monthly 109 (2002) pp. 277–278

[113] P.M. Rajković, M.D. Petković, P. Barry, *The Hankel Transform of the Sum of Consecutive Generalized Catalan Numbers*, Integral Transforms and Special Functions, Vol **18/4** (January 2007), 285 – 296.

[114] P. Robert, *On the Group inverse of a linear transformation*, J. Math. Anal. Appl. **22** (1968) 658–669.

[115] N.J. Rose, *A note on computing the Drazin inverse* Linear Algebra Appl. **15** (1976), 95–98.

[116] W. Rudin, *Real and complex analysis*, Third edition, McGraw-Hill, 1987.

[117] A. Schuster, P. Hippe, *Inversion of Polynomial Matrices by Interpolation* IEEE Transactions on Automatic control **37**, No. 3 (March 1992) 363–365.

[118] J.R. Sendra, *Hankel Matrices and Computer Algebra*, ACM SIGSAM Bulletin, Vol **24**, Issue 3, (July 1990), 17 – 26.

[119] N. Shinozaki, M. Sibuya, K. Tanabe, *Numerical algorithms for the Moore-Penrose inverse of a matrix: direct methods*, Annals of the Institute of Statistical Mathematics, **24** (1972) 193–203.

[120] T. Söderström, G. W. Stewart, *On the numerical properties of an iterative method for computing the Moore-Penrose generalized inverse*, SIAM J. Numer. Anal. **11** (1974) 61–74.

[121] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, http://www.research.att.com/∼njas/sequences/.

[122] M.Z. Spivey, L.L. Steil, *The k-Binomial Transforms and the Hankel Transform*, Journal of Integer Sequences, Vol. **9** (2006), Article 06.1.1.

[123] P.S. Stanimirović, *Programski paket MATHEMATICA i primene*, Elektronski fakultet, Niš, 2002.

[124] P.S. Stanimirović, *A finite algorithm for generalized inverses of polynomial and rational matrices*, Appl. Math. Comput. **144** (2003) 199–214.

[125] P.S. Stanimirović, *Block representation of* $\{2\}$, $\{1,2\}$ *inverses and the Drazin inverse*, Indian Journal Pure Appl. Math., **29** (1998) 1159–1176.

[126] P.S. Stanimirović, *General determinantal representation of generalized inverses over integral domains*, Publicationes Mathematicae Debrecen, **54** (1999), 221–249.

[127] P.S. Stanimirović, *Limit representations of generalized inverses and related methods*, Appl. Math. Comput. **103** (1999), 51–68.

[128] P.S. Stanimirović, *Computing pseudoinverses using minors of an arbitrary matrix*, Filomat 9:2 (1995), 285–294.

[129] P.S. Stanimirovic, N.P. Karampetakis, *Symbolic implementation of Leverrier-Faddeev algorithm and applications*, 8th IEEE Medit. Conference on Control and Automation, Patra, Greece, 2000.

[130] P.S. Stanimirović, M.D. Petković, *Computation of generalized inverses of polynomial matrices by interpolation*, Applied Mathematics and Computation, **172/1** (2006), 508–523.

[131] P.S. Stanimirović, N.V. Stojković, M.D. Petković, *Matematičko programiranje*, PMF Niš, 2007.

[132] P.S. Stanimirović, M.B. Tasić, *Partitioning method for rational and polynomial matrices*, Appl. Math. Comput., **155** (2004), 137–163.

[133] P.S. Stanimirović, M.B. Tasić, *A problem in computation of pseudoinverses*, Appl. Math. Comput., **135** (2-3) (2003), 443–469.

[134] P.S. Stanimirović, M.B. Tasić, *Drazin inverse of one-variable polynomial matrices*, Filomat, **15** (2001), 71–78.

[135] P.S. Stanimirović, M.B. Tasić, *Partitioning method for rational and polynomial matrices*, Appl. Math. Comput. **155** (2004) 137–163.

[136] P.S. Stanimirović, M.B. Tasić, *Computing generalized inverses using LU factorization of matrix product*, International Journal of Computer Mathematics, doi: 10.1080/00207160701582077.

[137] G.W. Stewart, *Introduction to Matrix Computation*, Academic Press, New York, 1973.

[138] V. Strassen, *Gaussian Elimination is Not Optimal*, Numerische Mathematik 13, 354-356, 1969.

[139] M. Stojić, *Kontinualni sistemi automatskog upravljanja*, Gradjevinska knjiga, Beograd, 1983.

[140] G. Szegö, *Orthogonal polynomials (Fourth ed.)*, Vol. **23** of AMS Colloquium Publications. Providence, RI: Amer. Math. Soc. 1975.

[141] M.B. Tasić, P.S. Stanimirović, M.D. Petković, *Symbolic computation of weighted Moore-Penrose inverse using partitioning method*, Applied Mathematics and Computation, **189** (2007) 615–640.

[142] R.P. Tewarson, *A direct method for generalized matrix inversion*, SIAM J. Numer. Anal. **4** (1967) 499–507.

[143] S. Toledo, *Locality of reference in LU decomposition with partial pivoting*, SIAM Journal of Matrix Analysis and Applications, **18** (1997), 1065–1081.

[144] P. Mladenović, *Kombinatorika*, Društvo Matematičara Srbije, Beograd 1992.

[145] X. Viennot, *Une theorie combinatoire des polynomes orthogonaux generaux*, UQAM, Montreal, Quebec, 1983.

[146] H.S. Wall, *Analytic Theory of Continued Fractions*, Van Nostrand, New York, 1948.

[147] G.R. Wang, *A new proof of Grevile's method for computing the weighted M-P inverse*, Journal of Shangai Normal University (Natural Science Edition), **3** 1985.

[148] G. Wang, Y. Wei, S. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, 2003.

[149] G.R. Wang, Y.L.Chen, *A recursive algorithm for computing the weighted Moore-Penrose inverse* $A^{\dagger}_{MN}$, Journal of Computational mathematics, **4** (1986), 74–85.

[150] Y. Wei, *A characterization and representation of the generalized inverse* $A^{(2)}_{T,S}$ *and its applications*, Linear Algebra Appl., **280** (1998), 87–96.

[151] Y. Wei, *Index splitting for the Drazin inverse and the singular linear system*, Appl. Math. Comput. **95** (1998) 115–124.

[152] Y. Wei, D. S. Djordjević, *On integral representation of the generalized inverse* $A^{(2)}_{T,S}$, Appl. Math. Comput., **142** (2003), 189–194.

[153] Y. Wei, H. Wu, *The representation and approximation for Drazin inverse*, J. Comput. Appl. Math., **126** (2000) 417–432.

[154] Y. Wei, H. Wu, *The representation and approximation for the generalized inverse* $A^{(2)}_{T,S}$, Appl. Math. Comput., **135** (2003), 263–276.

[155] Y. Wei, N. Zhang, *A note on the representation and approximation of the outer inverse* $A^{(2)}_{T,S}$ *of a matrix* A, Appl. Math. Comput., **147** (2004), no. 3, 837–841.

[156] W. J. Woan, *Hankel Matrices and Lattice Paths,* Journal of Integer Sequences, Article 01.1.2, Volume 4, 2001.

[157] S. Wolfram, *Mathematica Book, Version 3.0*, Wolfram Media and Cambridge University Press, 1996.

[158] S. Wolfram, *The Mathematica Book, 4th ed.*, Wolfram Media/Cambridge University Press, 1999.

[159] S. Wolfram, *Mathematica Book, Version 5.0*, Wolfram Media and Cambridge University Press, 1996.

[160] F. Zhang, *Matrix Theory: Basic Results and Techniques*, New York: Springer-Verlag, 1999.

[161] G. Zielke, *Motivation und Darstellung von verallgemeinerten Matrixinversen*, Beiträge zur Numerischen Mathematik, **7** (1979) 177–218.

[162] G. Zielke, *Report on test matrices for generalized inverses*, Computing **36** (1986) 105–162.

[163] E.L. Žukovski, R.S. Lipcer, *On recurent computation of normal solutions of linear algebraic equations*, Ž. Vicisl. Mat. i Mat. Fiz., **12** (1972) 843–857 (In Russian).

[164] E.L. Žukovski, R.S. Lipcer, *On computation pseudoinverse matrices*, Ž. Vicisl. Mat. i Mat. Fiz., **15** (1975) 489–492 (In Russian).

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:	
Идентификациони број, **ИБР**:	
Тип документације, **ТД**:	монографска
Тип записа, **ТЗ**:	текстуални / графички
Врста рада, **ВР**:	Докторска дисертација
Аутор, **АУ**:	Марко Д. Петковић
Ментор, **МН**:	Предраг С. Станимировић
Наслов рада, **НР**:	Симболичко израчунавање Ханкелових детерминанти и генералисаних инверза матрица
Језик публикације, **ЈП**:	српски
Језик извода, **ЈИ**:	српски
Земља публиковања, **ЗП**:	Србија
Уже географско подручје, **УГП**:	Србија
Година, **ГО**:	2008.
Издавач, **ИЗ**:	ауторски репринт
Место и адреса, **МА**:	Ниш, Вишеградска 33
Физички опис рада, **ФО**:	5 / 172 / 164 / 30 / 2 / 0
Научна област, **НО**:	рачунарске науке
Научна дисциплина, **НД**:	симболичко израчунавање
Предметна одредница/Кључне речи, **ПО**:	симболичко рачунање, Ханкелове детерминанте, генералисани инверзи
УДК	512.643
Чува се, **ЧУ**:	библиотека
Важна напомена, **ВН**:	
Извод, **ИЗ**:	У овој дисертацији су модификовани постојећи и уведени нови методи за симболичко рачунање Ханкелових детерминанти и генералисаних инверза матрица. Изведени су изрази за Ханкелову детерминанту за различите класе низова. Конструисан је метод за брзо рачунање генералисаних инверза који достиже теоријску доњу границу сложености. Такође, конструисано је неколико метода за рачунање генералисаних инверза рационалних и полиномијалних матрица.
Датум прихватања теме, **ДП**:	09.05.2008. године
Датум одбране, **ДО**:	
Чланови комисије, **КО**: Председник:	
Члан:	
Члан:	
Члан:	
Члан, ментор:	

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

KEY WORDS DOCUMENTATION

Accession number, **ANO**:	
Identification number, **INO**:	
Document type, **DT**:	monograph
Type of record, **TR**:	textual / graphic
Contents code, **CC**:	doctoral dissertation
Author, **AU**:	Marko D. Petković
Mentor, **MN**:	Predrag S. Stanimirović
Title, **TI**:	Symbolic computation of Hankel determinants and matrix generalized inverses
Language of text, **LT**:	Serbian
Language of abstract, **LA**:	English
Country of publication, **CP**:	Serbia
Locality of publication, **LP**:	Serbia
Publication year, **PY**:	2008
Publisher, **PB**:	author's reprint
Publication place, **PP**:	Niš, Višegradska 33
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)	5 / 172 / 164 / 30 / 2 / 0
Scientific field, **SF**:	computer science
Scientific discipline, **SD**:	symbolic computation
Subject/Key words, **S/KW**:	symbolic computation, hankel determinants, generalized inverses
UC	512.643
Holding data, **HD**:	library
Note, **N**:	
Abstract, **AB**:	In this thesis, existing methods for symbolic computation of Hankel deteriminants and matrix generalized inverses are modified and new are introducted. There are derived closed-form expressions for Hankel determinants of different classes of sequences. It is constructed the method for rapid computation of generalized inverses whose complexity reaches theoretical lower bound. There are also constructed new methods for computation of generalized inverses of rational and polynomial matrices.
Accepted by the Scientific Board on, **ASB**:	09.05.2008.
Defended on, **DE**:	
Defended Board, **DB**: President:	
Member:	
Member:	
Member:	
Member, Mentor:	