UNIVERZITET U NIŠU
PRIRODNO-MATEMATIČKI FAKULTET
Odsek za Matematiku

mr Milan B. Tasić

IZRAČUNAVANJE GENERALISANIH INVERZA

Doktorska disertacija

NIŠ, 2003

SADRŽAJ

PREDGOVOR

Istorijski posmatrano, još 1809. godine kod C.F.Gauss-a je implicitno sadržana ideja o generalisanim inverzima, i to u vezi sa uvođenjem principa metoda najmanjih kvadrata kod nekonzistentnih sistema. Iako se teorija generalisanih inverza u stvari razvila u poslednjih tridesetak godina, I.Fredholm je još 1903. godine definisao pseudoinverz linearnog integralnog operatora koji nije invertibilan u običnom smislu, dok je W.A.Hurwitz 1912. godine uveo pojam pseudorezolventnog operatora. Generalisani inverzi diferencijalnih operatora implicitno su sadržani u Hilbert-ovom razmatranju generalisane Green-ove funkcije 1904. godine, a kasnije su ih proučavali i drugi autori. E.H.Moore je prvi definisao i proučio jedinstveni generalisani inverz proizvoljne matrice, nazvavši ga "recipročnost matrice". Moguće je da je do ovih rezultata Moore došao još 1906. godine, mada su prvi objavljeni tek 1920. godine. Međutim njegov rad malo je bio poznat širokoj javnosti, verovatno zbog specifičnosti terminologije i oznaka. Tek 1955. godine rad R.Penrose-a [47], pobudio je pravi interes za izučavanje ove problematike.

Zajednički problem nastao pri implementaciji različitih metoda za numeričko izračunavanje generalisanih inverza je brzo uvećanje operacija u pokretnom zarezu. U rekurzivnoj implementaciji *partitioning metoda* [74], glavni problem je ponavljanje izračunavanja generalisanih inverza nekih matrica. U algoritmu za implementaciju limit reprezentacije generalisanih inverza, predloženom u [74], ponovna izračunavanja su angažovana u nekoliko izraza, koji koriste blokovske matrice sa identičnim blokovima. Konačno, implementacija *generalne determinantske reprezentacije* vodi ka multiplikativnom ponovnom izračunavanju nekih minora dve date matrice. Tim pre, ovaj metod zahteva takođe izračunavanje znatnog broja različitih minora.

U disertaciji su opisani odgovarajući algoritmi koji eliminišu ove poteškoće. Prvi pristup, koristi mogućnosti paketa MATHEMATICA verzija 4.0 (videti [83, 84]) u simboličkom izračunavanju:

1. Simboličke matrične algebre, koja uključuje simboličke operacije nad vektorima i matricama.

2. Algebarske manipulacije, koja uključuje izračunavanja sa simbolima i operacijama na algebarskim izrazima.

3. Ugrađena funkcija `Simplify[expr]`, koja obezbeđuje niz algebarskih manipulacija nad *expr*, a vraća *expr* u pojednostavljenoj formi.

Drugi pristup je baziran na konstrukciji nekoliko baza podataka u funkciji programskog paketa DELPHI [11].

Disertacija se sastoji iz tri glave. Prvi glava je uvodnog karaktera i u njoj su opisani osnovi pojmovi vezani za problem izračunavanja generalisanih inverza.

U ovoj glavi, su opisani neki poznati rezultati vezani za generalisane inverze kao i ukazani uočeni nedostaci metoda i algoritama za izračunavanje inverza. U drugoj glavi, koja predstavlja i najveći deo disertacije, opisani su metodi za simboličko izračunavanje generalisanih inverza. U ovoj glavi je izučavan problem simboličkog izračunavanja Moore-Penroseovog inverza na racionalne, polinomijalne i retko posednute matrice, a glava se sastoji iz tri poglavlja i to: generalisani inverzi numeričkih i racionalnih matrica [74], generalisani inverzi polinomijalnih matrica [75] i generalisani inverzi retko posednutih matrica [77]. Poglavlja sadraže nove algoritme, metode i mnoštvo numeričkih rezultata i test primera. Algoritmi koji su predloženi u mnogome poboljšavaju konvergenciju metoda, a primeri koji su dati opravdavaju njihovo uvođenje. Algoritmi su naročito pogodni za primenu u proceduralnim programskim jezicima, koji nemaju mogućnost simboličkog izračunavanja. Ova glava sadrži i još neobjavljenih rezultata, kao što su: efektivni algoritam za izračunavanje Drazinovog inverza sa jednom i više promenljivih, metoda pregrađivanja za polinomijalne matrice, metoda inverzije interpolacijom itd. U trećoj glavi, je opisan metod za iračunavanje generalisanih inverza pomoću baza podataka, koji opravdava uvođenje sličnih algoritama u proceduralnim programskim jezicima. Metoda determinantske reprezentacije vodi ponovnom izračunavanju istih minora za dve date matrice. Algoritam zahteva izračunavanja značajnog broja različitih minora. Pristup, koji omogućuje, da se izbegnu ponovna izračunavanja istih minora, bazira se na konstrukciji nekoliko baza podataka. U bazama se čuvaju rezultati i svi među rezultati. U izrazu koji koristi ranije izračunate minore potrebno je samo pročitati njegovu vrednost iz odgovarajuće baze podataka u koju je smešten [74].

Napomenimo da je ova disertacija zasnivala pre svega na originalnim objavljenim i još neobjavljenim rezultatima, a predložena rešenja problema mogu da se iskoriste u različitim problemima i situacijama. Aktuelnost ovako izabrane disertacije, najbolje se vidi u velikom broju radova, koji se na tu temu objavljuju zadnjih godina. Problematika vezana za temu disertacije, po American Mathematics Subject Classification (1991) nalazi se primarno u: 15A09 i 68Q40. Svi kodovi programa i test primeri, u disertaciji su u izvornom obliku i mogu se po potrebi koristiti u sličnim problemima.

Želim da izrazim veliku zahvalnost mom mentoru dr Predragu Stanimiroviću, profesoru Prirodno-matematičkog fakulteta u Nišu, na korisnim savetima, pravilnom usmerenju i nesebičnoj pomoći.

Najveću zahvalnost dugujem mojoj porodici na ljubavi, strpljenju i podršci.

U Nišu 2003. godine.

1 UVOD

Generalisani inverzi proučavaju se u mnogim matematičkim disciplinama, npr. u linearnoj algebri, teoriji operatora, teoriji semigrupa, prstena, itd. Potreba za jednom takvom teorijom javila se u vezi tzv. nekorektno postavljenih linearnih problema. Prema J.Hadamard-u (1902), jednačina $Ax = y$, gde je A preslikavanje iz topološkog prostora X u topološki prostor Y predstavlja korektno postavljen problem ako:

a) rešenje $x \in X$ ove jednačine postoji za svako $y \in Y$;

b) rešenje $x \in X$ je jedinstveno u X;

c) postoji neprekidna zavisnost x od y.

Problem je korektno postavljen ako postoji inverzno preslikavanje $A^{-1} : Y \to X$ i ako je ono neprekidno. Na primer, ako je A nesingularna matrica, postoji jedinstvena inverzna matrica A^{-1}, tako da je $AA^{-1} = A^{-1}A = I$. U tom slučaju sistem linearnih jednačina $Ax = y$ ima jedinstveno rešenje $x = A^{-1}y$. Međutim, ako je A singularna ili pravougaona, A^{-1} ne postoji, ali to ne znači da rešenje sistema ne postoji. Čak i kod sistema koji nisu konzistentni, može se posmatrati približno rešenje dobijeno metodom najmanjih kvadrata. Ove probleme, kao i mnoge druge u numeričkoj algebri, optimizaciji i sistemima upravljanja, teoriji igara i električnih kola, programiranju, statistici, ekonomiji i drugim oblastima, moguće je rešiti uvođenjem generalisanih inverza matrice ili linearnog operatora. Da bi generalisani inverz, kao uopštenje običnog inverza, bio od koristi u rešavanju ovih problema, treba da ispunjava sledeće uslove: da postoji za širu klasu operatora od klase invertibilnih operatora, da ima neka svojstva običnog inverza i da se u slučaju invertibilnog operatora svodi na običan inverz.

Moore-Penroseov inverz

Poznat je veći broj ekvivalentnih definicija Moore-Penroseovog inverza. R. Penrose je 1955. godine dokazao sledeću teoremu [47]:

Teorema 1.1 (Penrose) *Za datu matricu* $A \in \mathbf{C}^{m,n}$ *postoji jedinstvena matrica* $X \in \mathbb{C}^{n,m}$ *koja ispunjava jednačine*

$$\begin{array}{llll} (1) & AXA = A & (2) & XAX = X \\ (3) & (AX)^* = AX & (4) & (XA)^* = XA \end{array}$$

Penrose je matricu X označio sa $A^\dagger$ i nazvao je *generalisani inverz matrice* A. On postoji za singularne i pravougaone matrice, a u slučaju regularne matrice je $A^\dagger = A^{-1}$. Za matricu $A^\dagger$ koristi se naziv Moore-Penroseov inverz matrice A.

Teorema 1.2 (Moore) *Za datu matricu $A \in \mathbb{C}^{m,n}$ postoji jedinstvena matrica $X \in \mathbb{C}^{n\times m}$ tako da za pogodno izabrane matrice Y i Z važi:*

$$AXA = A, \qquad X = YA^* = A^*Z.$$

Osim toga je

$$XAX = X, \qquad (AX)^* = AX, \qquad (XA)^* = XA.$$

Definicija 1.1 (Funkcionalna definicija generalisanog inverza) *Za datu matricu $A \in \mathbb{C}^{m\times n}$ definišimo linearnu transformaciju $\tilde{A}^\dagger : \mathbb{C}^m \to \mathbb{C}^n$ relacijom $\tilde{A}^\dagger x = 0$ ako $x \in R(A)^\perp$ i $\tilde{A}^\dagger x = \left(\tilde{A}_{|_{R(A^*)}}\right)^{-1} x$ ako $x \in R(A)$. Matrica linearne transformacije $\tilde{A}^\dagger$ označava se sa $A^\dagger$ i naziva se generalisani inverz za A.*

Definicija 1.2 (Mooreova definicija) *Generalisani inverz za $A \in \mathbb{C}^{m\times n}$ je jedinstvena matrica $A^\dagger$ takva da je*

$$\text{(i)} \quad \mathrm{AA^\dagger} = \mathrm{P_{R(A)}}, \quad \text{(ii)} \quad \mathrm{A^\dagger A} = \mathrm{P_{R(A^\dagger)}}.$$

Definicija 1.3 [47] (Penroseova definicija). *Za $A \in \mathbb{C}^{m\times n}$ generalisani inverz je jedinstvena matrica $A^\dagger \in \mathbb{C}^{n\times m}$ koja ispunjava jednačine* (1), (2), (3) *i* (4).

Teorema 1.3 *Funkcionalna, Mooreova i Penroseova definicija generalisanog inverza su ekvivalentne.*

Teorema 1.4 [33] *Neka je $A = PQ$ potpuna rang faktorizacija matrice A, tj. $P \in \mathbb{C}_r^{m\times r}$ i $Q \in \mathbb{C}_r^{r\times n}$. Tada je*

$$A^\dagger = Q^\dagger P^\dagger = Q^*(QQ^*)^{-1}(P^*P)^{-1}P^*.$$

Aproksimativna svojstva generalisanih inverza.

Posmatrajmo problem rešavanja linearnog sistema $Ax = b$, gde su $A \in \mathbb{C}^{m\times n}$ i $b \in \mathbb{C}^m$ zadati. Ovaj problem ima rešenje ako i samo ako je $b \in R(A)$; rešenje je jedinstveno ako i samo ako je $N(A) = \{0\}$. Osim toga, posmatra se sledeći problem: ako $b \notin R(A)$, tada odrediti x, tako da Ax bude "najbliže" vektoru b.

Definicija 1.4 [33], [54] *Dat je sistem jednačina $Ax = b$, $A \in \mathbb{C}^{m\times n}$. Pseudoinverz X je minimalne norme matrice A ako je za svako $b \in R(A)$, $x = Xb$ rešenje jednačine $Ax = b$, i ako je*

$$\min_{Ax=b} \|x\| = \|Xb\|.$$

Teorema 1.5 [33], [54] *X je g-inverz matrice A takav da je Xb rešenje sistema $Ax = b$ minimalne norme ako i samo ako X ispunjava uslove*

$$AXA = A \qquad\qquad (XA)^* = XA.$$

Lema 1.1 [54] *Ako je $\|x\|^2 = x^*Nx$ (ili $\langle x, y\rangle = y^*Nx$), gde je N pozitivna matrica, tada možemo pisati*

$$AXA = A \qquad\qquad (XA)^* = NXA.$$

Teorema 1.6 [33] *Neka je $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$. Ako je sistem $Ax = b$ konzistentan, vektor $x = A^{(1,4)}b$, $A^{(1,4)} \in A\{1, 4\}$ je jedinstveno rešenje za koje je $\|x\|$ najmanja. Važi i obrnuto, tj. ako je matrica $X \in \mathbb{C}^{n\times m}$ takva da u slučaju konzistentnog sistema $Ax = y$, $x = Xb$ predstavlja rešenje sa najmanjom normom, tada $X \in A\{1, 4\}$.*

Lema 1.2 *Rešenje sa minimalnom normom je jedinstveno (mada ne mora biti i g-inverzija jedinstvena).*

Teorema 1.7 [54] *Neka je M pozitivno semi-definitna matrica i neka je X g-inverz od A, takav da je Xb rešenje od $Ax = b$ sa minimalnom semi-normom. Tada A zadovoljava uslove:*

$$AXA = A \qquad (XA)^*M = MXA.$$

Definicija 1.5 [54] *Data je nekonzistentna jednačina $Ax = b$. Vektor x_0 je najmanje kvadratno rešenje ako je*

$$\|Ax_0 - b\| \leq \|Ax - b\|, \quad (x \in C^n).$$

X je težinski najmanje srednje-kvadratni inverz za A sa težinom M ako za svako $b \in \mathbb{C}^m$ vektor $x = Xb$ obezbeđuje minimalnu vrednost izraza

$$\|Ax - b\|_M\,, \quad \forall x \in \mathbb{C}^n.$$

Za pseudoinverz X koristi se oznaka $A_{l(M)}$, ili A_l, dok se klasa takvih inverza označava sa $\{A_{l(M)}\}$, ili sa $\{A_l\}$. Podklase ovih klasa u kojima su izdvojena rešenja jednačine $Ax = b$ označavaju se sa $\{A^-_{l(M)}\}$, odnosno $\{A^-_l\}$.

Teorema 1.8 [54] *X je g-inverz od A, takav da je Xb najmanje srednje-kvadratno rešenje jednačine $Ax = b$, za svako $b \in \mathbb{C}^m$, ako i samo ako X ispunjava uslove*

$$AXA = A \qquad (AX)^* = AX.$$

Teorema 1.9 [33] *Za $A \in \mathbb{C}^{m\times n}$ i $b \in \mathbb{C}^m$, $\|Ax - b\|$ je najmanje za $x = A^{(1,3)}b$. Obrnuto, ako je $X \in \mathbb{C}^{n\times m}$ takva matrica da je za svaki vektor b norma $\|Ax - b\|$ najmanja, tada $X \in A\{1, 3\}$.*

Ako je Xb najmanje srednje-kvadratno rešenje, tada je klasa najmanje srednje-kvadratnih rešenja data sa

$$Xb + (I - XA)z, \quad z \quad \text{proizvoljno.}$$

Pseudoinverz G koja obezbeđuje najmanje srednje-kvadratno rešenje sistema $Ax = b$ se označava sa A_l^- kada je norma indukovana skalarnim proizvodom $\langle \cdot, \cdot \rangle$ ili sa $A_{l(M)}^-$, u slučaju kada je norma izvedena iz skalarnog proizvoda $\langle \cdot, \cdot \rangle_M$ (i naziva M-najmanje srednje-kvadratni g-inverz).

Veza između Moore-Penroseovog inverza i najmanje-kvadratnog rešenja minimalne norme, koju je prvi dokazao R. Penrose [47] data je sledećom teoremom.

Teorema 1.10 *Neka je $A \in \mathbb{C}^{m\times n}$ i $b \in \mathbb{C}^m$. Tada, između svih najmanje-kvadratnih rešenja jednačine $Ax = b$, minimalne norme je $x = A^\dagger b$. Obrnuto, ako je $X \in \mathbb{C}^{n\times m}$ takva matrica da je za svaki vektor b norma Xb najmanje-kvadratno rešenje minimalne norme, tada je $X = A^\dagger$.*

Lema 1.3 [54] *Ako je $\|y\| = \|y\|_M = (y^*My)^{\frac{1}{2}}$ i $\|x\|_n = \|x\|_N = (x^*Nx)^{\frac{1}{2}}$, gde su M i N pozitivno definitne, tada X ispunjava jednačine*

$$AXA = A,\ XAX = X,\ (AX)^*M = MAX\ (XA)^*N = NXA$$

Rao i Mitra [54] su g-inverz koji zadovoljava uslove Leme označili sa $A_{M,N}^\dagger$, što predstavlja rešenje sa minimalnom N-normom (ili seminormom) i najmanje kvadratnom M-normom. Kada su M i N jedinične matrice, koristimo oznaku $A^\dagger$ izostavljajući indekse.

Ben Israel i Grevile [33] su koristili nešto drugačije oznake, i dokazali opštiju teoremu.

Teorema 1.11 *Neka su $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$, i neka su $M \in \mathbb{C}^{m\times m}$, $N \in \mathbb{C}^{n\times n}$ pozitivno definitne matrice. Tada postoji jedinstvena matrica $X = A_{(M,N)}^\dagger \in A\{1, 2\}$ koja ispunjava uslove*

$$(MAX)^* = MAX\,, \qquad (NXA)^* = NXA.$$

Osim toga, $\|Ax-b\|_M$ uzima minimalnu vrednost za $x = Xb$, i za skup vektora x rešenje $x = Xb$ ima minimalnu normu $\|x\|_N$.
Obrnuto, ako $Y \in \mathbb{C}^{n\times m}$ ima osobinu da je, za sve b, $x = Yb$, takav vektor za koji je $\|x\|_N$ najmanje i $\|Ax-b\|_M$ minimalno, tada je $Y = A^{(1,2)}_{(M,N)}$.

Lema 1.4 *$A^{\dagger}_{M,N}$ je jedinstveno ako je M pozitivno definitna. Kada su N i M pozitivno semidefinitne tada $A^{\dagger}_{M,N}$ ne mora biti jedinstveno.*

Teorema 1.12 [54] *Svaka matrica $A \in \mathbb{C}^{m\times n}_r$ može se predstaviti u obliku proizvoda $A = PQ$ dve matrice potpunog ranga $P \in \mathbb{C}^{m\times r}_r$ i $Q \in \mathbb{C}^{r\times n}_r$.*

Ovakva dekompozicija nije jedinstvena. Međutim, ako je jedna od matrica P ili Q zadata, tada je druga matrica jednoznačno određena. Potpuna rang faktorizacija se može upotrebiti za izračunavanje različitih klasa generalisanih inverza.

Teorema 1.13 *Ako $A \in \mathbb{C}^{m\times n}_r$ ima potpunu rang faktorizaciju $A = PQ$, ($P \in \mathbb{C}^{m\times r}_r$, ($Q \in \mathbb{C}^{r\times n}_r$), i ako su $W_1 \in \mathbb{C}^{n\times r}$ i $W_2 \in \mathbb{C}^{r\times m}$ matrice koje ispunjavaju uslove rank(QW_1) = rank(W_2P) = rank(A), tada važi:*

$$\begin{aligned}
A^{\dagger} &= Q^*(QQ^*)^{-1}(P^*P)^{-1}P^* = Q^{\dagger}P^{\dagger} \quad [5]\\
A\{1,2\} &= W_1(QW_1)^{-1}(W_2P)^{-1}W_2 = Q_r^{-1}P_l^{-1} \quad [54]\\
A\{1,2,3\} &= W_1(QW_1)^{-1}(P^*P)^{-1}P^* = Q_r^{-1}P^{\dagger} \quad [54]\\
A\{1,2,4\} &= Q^*(QQ^*)^{-1}(W_2P)^{-1}W_2 = Q^{\dagger}P_l^{-1} \quad [54]\\
A^{\#} &= P(QP)^{-2}Q,
\end{aligned}$$

gde P_l^{-1} predstavlja levi inverz matrice P, a Q_r^{-1} desni inverz matrice Q [43, 44].

Ovaj metod je naročito pogodan za matrice potpunog ranga, jer se tada jedna od matrica P i Q svodi na jediničnu. Na primer, za Moore-Penroseov inverz važi:

$$A^{\dagger} = \begin{cases} A^*(AA^*)^{-1}, & m < n \\ (A^*A)^{-1}A^*, & m > n. \end{cases}$$

Aproksimativna svojstva generalisanih inverza mogu se proučiti u [1, 5, 47]. Aproksimativna svojstva generalisanih inverza omogućavaju da se oni mogu izračunati minimizacijom dve norme, odnosno funkcije. Ova njihova korisna osobina omogućava da se poveže izračunavanje generalisanih inverza kao i LSS, $NLSS$ rešenja sa različitim metodima optimizacije.

Implementacija različitih metoda za izračunavanje LSS i $NLSS$, koji nisu bazirani na optimizaciji, može se naći u [29, 28, 49, 61].

Grevile je u [33] predložio rekurzivni algoritam koji daje Moore-Penroseov pseudoinverz matrice R proširene odgovarajućim vektorom r označenog sa $R^\dagger$. Algoritam je međutim, kao i mnogi numerički algoritmi za izračunavanje inverza, nestabilan. Dobro je poznato da je prisutna greška u zaokruživanju i neke male vrednosti uzimamo kao nula. Stoga je jasno da ove diskontinuitete treba izbeći. Tokom simboličke implementacije, promenljive se ostavljaju u tačnoj formi ili mogu biti "izostavljene" (bez numeričkih vrednosti), tako da nema gubitka koji može nastati zaokruživanjem.

Izračunavanje Moore-Penroseovog inverza za polinomijalne i racionalne matrice, baziran na Leverrier-Faddeevom algoritmu proučavan je u [2, 17, 22, 23, 24, 25]. U literaturi je poznat veći broj aplikacija za izračunavanje generisanje inverza polinomijalnih matrica [22, 23, 24, 25, 30, 31, 32]. U [22] opisana je implementacija algoritma za izračunavanje Moore-Penroseovog inverza singularnih racionalnih matrica u programskom jeziku MAPLE. Algoritmi koji koriste reprezentaciju matrice u polinomijalnoj formi, su naročito pogodni za implementaciju u programskim paketima koji ne mogu vršiti simbolička izračunavanja.

2 SIMBOLIČKA IMPLEMENTACIJA GENERALISANIH INVERZA

Pomoću direktnih metoda svaki element generalisanog inverza izračunava se direktno, bez iterativnih poboljšanja. Razmatra se implementacija sledećih grupa metoda:

1. metode bazirane na matričnim faktorizacijama [43];
2. metode koje proističu iz blokovskih dekompozicija [42, 44, 69, 70];
3. metode pregrađivanja [66];
4. determinantska reprezentacija [68].

Iterativnim metodama se rezultat prilikom izračunavanja generalisanog inverza u svakom iterativnom koraku poboljšava. Metode se mogu proučiti u [1, 5, 47].

Implementacija različitih metoda za izračunavanje LSS i $NLSS$, koji nisu bazirani na optimizaciji, mogu se naći u [29, 28, 49, 61].

O generalisanim inverzima i graničnim procesima može se pročitati u [63, 74].

2.1 INVERZI NUMERIČKIH I RACIONALNIH MATRICA

2.1.1 Elementarne matrične transformacije

Elementarne transformacije nad vrstama (kolonama) matrice $A \in \mathbb{C}_r^{m\times n}$ su:

1. zamena dve vrste (ili kolone) u A;
2. množenje svih elemenata vrste (ili kolone) matrice A nenula brojem;
3. sabiranje bilo koje vrste (ili kolone) u A sa nekom vrstom (ili kolonom) iz A, koja je prethodno pomnožena nekim brojem.

Može se pokazati da su ovakve transformacije nad vrstama matrice A ekvivalentne sa množenjem matrice A sa leve strane, odgovarajućim regularnim matricama. Analogno, elementarne transformacije nad kolonama se mogu zameniti množenjem regularnim matricama sa desne strane. Na primer, zamena vrste i_1 i i_2 ($i_1 < i_2$) matrice A je ekvivalentna množenju matrice A sa leve strane matricom $P(i_1, i_2)$, koja se dobija zamenom i_1 vrste i i_2 vrste u jediničnoj $m \times m$ matrici, sleva:

$$
P(i_1,i_2) = \begin{array}{r} \\ \\ \\ i_1 \rightarrow \\ \\ \\ \\ i_2 \rightarrow \\ \\ \\ \\ \end{array}
\begin{bmatrix}
1 & & & & & & & & & & \\
 & \ddots & & & & & & & & & \\
 & & 1 & & & & & & & & \\
 & & & 0 & \cdots & \cdots & \cdots & 1 & & & \\
 & & & & 1 & & & & & & \\
 & & & \vdots & & \ddots & & \vdots & & & \\
 & & & & & & 1 & & & & \\
 & & & 1 & \cdots & \cdots & \cdots & 0 & & & \\
 & & & & & & & & 1 & & \\
 & & & & & & & & & \ddots & \\
 & & & & & & & & & & 1
\end{bmatrix}.
$$

Slično, efekat množenja i-te vrste matrice A nenula elementom k se može postići formiranjem proizvoda $M(i,k)$, gde je matrica $M(i,k)$ dobijena iz jedinične matrice reda $m \times m$ množenjem njene i-te vrste sa k:

$$
M(i,k) = \begin{array}{r} \\ \\ \\ i \rightarrow \\ \\ \\ \\ \end{array}
\begin{bmatrix}
1 & & & & & & \\
 & \ddots & & & & & \\
 & & 1 & & & & \\
 & & & k & & & \\
 & & & & 1 & & \\
 & & & & & \ddots & \\
 & & & & & & 1
\end{bmatrix}.
$$

Konačno, dodavanje i_1 vrsti vrste i_2 pomnožene realnim brojem k je ekvivalentno množenju matrice A matricom $D(i_1,i_2,i_3)$ sleva, koja se dobija izvršenjem analognih transformacija na jediničnoj matrici reda m.

$$D(i_1, i_2, i_3) = \begin{cases} \begin{array}{c} \\ \\ i_1 \rightarrow \\ \\ i_2 \rightarrow \\ \\ \\ \end{array} \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & \dots & k & & \\ & & & \ddots & \vdots & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}, & i_1 < i_2, \\ \begin{array}{c} \\ \\ i_2 \rightarrow \\ \\ i_1 \rightarrow \\ \\ \\ \end{array} \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & & \\ & & \vdots & \ddots & & & \\ & & k & \dots & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}, & i_2 < i_1. \end{cases}$$

Množenje sdesna matrice A matricama $P(i_1, i_2)$, $M(i, k)$, $D(i_1, i_2, i_3)$, odgovarajućih dimenzija, ekvivalentno je odgovarajućim promenama na njenim kolonama. Elementarnim transformacijama se matrica može redukovati u jednostavniji oblik koji sadrži veći broj nula od polazne matrice.

Primer 2.1 Zamena druge i četvrte vrste matrice A vrši se množenjem sa leve strane matricom

$$P(2,4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Zamena prve i treće kolone matrice A postiže se matričnim proizvodom $AP(1,3)$, gde je

$$P(1,3) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Množenje treće vrste matrice A brojem 10 može se postići proizvodom $M(3,10)A$, gde je

$$M(3,10) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Dodavanje četvrte vrste matrice A pomnožene sa 2, njenoj drugoj vrsti, ekvivalentno je matričnom proizvodu $D(4,2,2)A$, za

$$D(4,2,2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Sledi implementacija elementarnih matričnih transformacija.

```
ChRows[a_,i1_,i2_]:=
  Block[{b=a,m,e,f},
      m=Dimensions[b][[1]];    e=IdentityMatrix[m];
      f=e[[i1]];e[[i1]]=e[[i2]];e[[i2]]=f;
      e.b
  ]; MatrixQ[a]

ChColumns[a_,i1_,i2_]:=
  Block[{b=a,n,e,f},
      n=Dimensions[b][[2]];    e=IdentityMatrix[n];
      f=e[[i1]];e[[i1]]=e[[i2]];e[[i2]]=f;
      b.e
  ];MatrixQ[a]

MultRow[a_,i_,k_]:=
  Block[{b=a,m,e},
      m=Dimensions[b][[1]];    e=IdentityMatrix[m];
      e[[i]]=k e[[i]];
      e.b
  ];MatrixQ[a]

MultColumn[a_,i_,k_]:=
  Block[{b=a,m,e},
      n=Dimensions[b][[2]];    e=IdentityMatrix[n];
      e[[i]]=k e[[i]];
      b.e
  ];MatrixQ[a]

AddRow[a_,i1_,i2_,k_]:=
  Block[{b=a,m,e,f},
      m=Dimensions[b][[1]];    e=IdentityMatrix[m];
      f=e[[i1]];f[[i2]]=k;e[[i1]]=f;
      e.b
  ];MatrixQ[a]
```

```
AddColumn[a_,i1_,i2_,k_]:=
  Block[{b=a,d,n,e,f},
      d=Dimensions[b];n=d[[2]];
      e=IdentityMatrix[n];
      f=e[[i2]];f[[i1]]=k;   e[[i2]]=f;
      b.e
  ];MatrixQ[a]
```

2.1.2 Izračunavanje pomoću potpune rang faktorizacije

Potpuna rang faktorizacija se može odrediti na više načina. U tu svrhu se može upotrebiti bilo koja od blokovskih reprezentacija matrica, o čemu će biti reči.

Poznat je sledeći metod za potpunu rang faktorizaciju matrice A:

Korak 1. Gaussovim metodom eliminacije naći hermitsku formu H_A matrice A.

Korak 2. Matrica P je sačinjena od onih kolona matrice A koje odgovaraju jediničnim kolonama hermitske forme H_A.

Korak 3. Matricu Q čine linearno nezavisne vrste matrice H_A (nenula vrste H_A).

Kao jedan od metoda potpune rang faktorizacije matrica može se uzeti LU dekompozicija.

Stav 2.1 *Svaka matrica $A \in \mathbb{C}_r^{m\times n}$ se može transformisati u oblik $A = LU$, gde je L donja trougaona matrica sa jedinicama na glavnoj dijagonali, a U je gornja trougaona matrica.*

Dokaz. LU faktorizacija se može izvesti iz Gaussovog metoda eliminacije sa potpunim izborom glavnog elementa. Matrica $A = A_0 \in \mathbb{C}_r^{m\times n}$ se transformiše redom u matrice $A_1, \ldots, A_r$. Matrica A_k, $0 \le k \le r$ jeste $m \times n$ matrica oblika

$$A_k = \left(a_{ij}^{(k)}\right) = \begin{bmatrix} U_k & V_k \\ \mathbb{O} & W_k \end{bmatrix},$$

gde je U_k trougaona matrica dimenzija $k \times k$. Označimo (i, j)-element u bloku W_k sa w_{ij}. Ako je $w_{\alpha\beta}$ najveći po modulu element u bloku W_k vrši se zamena α vrste sa $k + 1$ vrstom i β vrste sa $k + 1$ kolonom matrice A_k. Izvršenjem analognih zamena nad vrstama jedinične $m\times m$ matrice dobija se permutaciona matrica $E_k = P(k + 1, \alpha)$, dok se analognom zamenom kolona jedinične $n \times n$

matrice dobija permutaciona matrica $F_k = P(k+1,\beta)$. Uz pretpostavku $a_{ij}^{(1)} = a_{ij}, \quad 1 \le i \le m,\ 1 \le j \le n$, Gaussova eleminacija matrice A_k jeste sledeća transformacija:

$$\begin{aligned} v_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - v_{ik} a_{kj}^{(k)}, \quad \begin{pmatrix} i = k+1,\dots,m \\ j = k+1,\dots,n \end{pmatrix}, \quad k = 1,\dots,r. \end{aligned}$$

Kao rezultat ove transformacije dobija se matrica

$$A_{k+1} = \begin{bmatrix} U_{k+1} & V_{k+1} \\ \mathbb{O} & W_{k+1} \end{bmatrix}.$$

Posle r koraka dobija se matrica

$$A_r = \begin{bmatrix} U_r & V_r \\ \mathbb{O} & W_r \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1r}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2r}^{(2)} & \dots & a_{2n}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{rr}^{(r)} & \dots & a_{rn}^{(r)} \end{bmatrix}.$$

Za matricu U se može uzeti prvih r vrsta matrice A_r.

Matrica L je definisana sa

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ v_{21} & 1 & \dots & 0 & 0 \\ v_{31} & v_{32} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ v_{r1} & v_{r2} & \dots & v_{r,r-1} & 1 \\ \dots & \dots & \dots & \dots & \dots \\ v_{m1} & v_{m2} & \dots & v_{m,r-1} & v_{mr} \end{bmatrix}.$$

Može se izvršiti generalizacija postupka datog u [44], koja se sastoji u popunjavanju odgovarajućih mesta u matrici A dobijenim vrednostima. Na kraju transformacije, dobijamo:

$$EAF = LU \Longleftrightarrow A = E^* LUF^*,$$

Gde su E i F permutacione matrice definisane kao proizvod elementarnih matrica: $E = E_1 \cdots E_r, \quad F = F_1 \cdots F_r$. $\square$

Teorema 2.1 *Ako $A = LU$ predstavlja LU faktorizaciju matrice A, tada je*

$$A^{\dagger} = U^{\dagger}L^{\dagger} = U^{*}(UU^{*})^{-1}(L^{*}L)^{-1}L^{*}.$$

Standardna MATHEMATICA funkcija koja predstavlja LU faktorizaciju matrice A je $LUDecomposition[]$.

Primer 2.2 Za matricu $A = \begin{bmatrix} 1 & 3 & 6 & 9 \\ 1 & 2 & 3 & 4 \\ 0 & -2 & 0 & -2 \\ -1 & 0 & -1 & -3 \end{bmatrix}$ dobija se :

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{4}{9} & 1 & 0 & 0 \\ \frac{-2}{9} & 4 & 1 & 0 \\ \frac{-1}{3} & 3 & \frac{1}{4} & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & 3 & 6 & 9 \\ \frac{5}{9} & \frac{2}{3} & \frac{1}{3} & 0 \\ -2 & -4 & 0 & 0 \\ \frac{-11}{6} & 0 & 0 & 0 \end{bmatrix}$$

Koristi se veći broj direktnih metoda za izračunavanje Moore-Penroseovog inverza koji se zasnivaju na drugim matričnim faktorizacijama. Često se koristi QR faktorizacija matrica.

Teorema 2.2 *Ako je $A \in \mathbb{R}_r^{m\times n}$, tada postoji ortogonalna matrica $Q \in \mathbb{R}^{m\times n}$ i gornja trougaona $R \in \mathbb{R}^{n\times n}$, takve da je $A = QR$.*

Dokaz. Prvo izvodimo dokaz za regularne matrice [16]. Posmatraju se matrice rotacije oblika

$$T_{ij} = \begin{array}{r} \\ \\ \\ \\ i \rightarrow \\ \\ \\ j \rightarrow \\ \\ \\ \\ \end{array} \begin{bmatrix} 1 & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & c & \dots & \dots & \dots & -s & & & \\ & & & & 1 & & & & & & \\ & & & \vdots & & \ddots & & \vdots & & & \\ & & & & & & 1 & & & & \\ & & & s & \dots & \dots & \dots & c & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{bmatrix},$$

uz uslov $c^2 + s^2 = 1$. Tada postoji ugao θ takav da je $c = cos(\theta)$, $s = sin(\theta)$. Ako se matrica $A = (a_{\alpha,\beta})$ pomnoži sleva matricom T_{ij}, menjaju se samo i-ta i j-ta vrsta matrice A. Tačnije, za matricu $T_{ij}A = A^{(1)} = \left(a^{(1)}_{\alpha,\beta}\right)$ važi:

$$\begin{aligned} a^{(1)}_{i,\beta} &= ca_{i,\beta} - sa_{j,\beta}, \\ a^{(1)}_{j,\beta} &= sa_{i,\beta} + ca_{j,\beta}, \\ a^{(1)}_{\alpha,\beta} &= a_{\alpha,\beta}, \quad \alpha \notin \{i,j\}, \quad (\beta = 1,\dots,n.) \end{aligned}$$

Analogno, pri množenju matrice A sdesna matricom T_{ij} menjaju se samo i-ta i j-ta kolona, prema formulama:

$$\begin{aligned} a^{(1)}_{\alpha,i} &= ca_{\alpha,i} + sa_{\alpha,j}, \\ a^{(1)}_{\alpha,j} &= -sa_{\alpha,i} + ca_{\alpha,j}, \\ a^{(1)}_{\alpha,\beta} &= a_{\alpha,\beta}, \quad \beta \notin \{i,j\}, \quad (\alpha = 1,\dots,n.) \end{aligned}$$

Ako je bar jedan od elemenata $a_{i,\beta}$ i $a_{j,\beta}$ različit od nule, mogu se odabrati c i s tako da za matricu $A_{(1)} = T_{ij}A$ važi $a^{(1)}_{j,\beta} = 0$. To se postiže sa

$$s = -\frac{a_{j,\beta}}{\sqrt{a^2_{i,\beta}a^2_{j,\beta}}}, \qquad c = \frac{a_{i,\beta}}{\sqrt{a^2_{i,\beta}a^2_{j,\beta}}}.$$

Tada je

$$a^{(1)}_{i,\beta} = \sqrt{a^2_{i,\beta}a^2_{j,\beta}}, \qquad a^{(1)}_{j,\beta} = 0.$$

Ako je $a_{11} \neq 0$ matrica A se množi sleva matricama T_{12}, $T_{13}, \dots, T_{1n}$, tako da se uzastopno anuliraju svi elementi prve kolone osim a_{11}, koji će biti pozitivan posle svake transformacije. Ako je $a_{11} = 0$, postoji najmanji broj j takav da je $a_{j1} \neq 0$ (zbog regularnosti matrice A). Tada se matrica A množi sleva matricom T_{1j}. Posle transfromacije prve kolone, dobija se

$$A^{(1)} = T_{1n}T_{1,n-1}\cdots T_{12}A = \begin{bmatrix} a^{(1)}_{11} & a^{(1)}_{12} & \dots & a^{(1)}_{1n} \\ 0 & a^{(1)}_{22} & \dots & a^{(1)}_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & a^{(1)}_{n2} & \dots & a^{(1)}_{nn} \end{bmatrix}, \quad a^{(1)}_{11} > 0.$$

Ponavljajući ovakav postupak za ostale kolone, dobija se

$$A^{(n-1)} = \prod_{i=1}^{n-1}\prod_{k=i+1}^{n} T_{ik}A = \begin{bmatrix} a^{(1)}_{11} & a^{(1)}_{12} & \dots & a^{(1)}_{1n} \\ 0 & a^{(2)}_{22} & \dots & a^{(2)}_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a^{(n-1)}_{nn} \end{bmatrix}.$$

U matrici $A_{(n-1)}$ su svi dijagonalni elementi, osim možda $a_{nn}^{(n-1)}$ pozitivni. Koristeći oznaku $\prod_{i=1}^{n-1} \prod_{k=i+1}^{n} T_{ik} = Q^*$, dobijamo

$$A_{(n-1)} = R = Q^*A \Leftrightarrow A = QR.$$

Može se dogoditi da na i-tom koraku $(i < r)$ bude $a_{ji} = 0, \quad j = i+1, \dots, n$. Tada je potrebno uraditi sledeće: naći prvu nenultu kolonu. Neka je to k-ta kolona $(k > i)$. Izvršiti zamenu k-te i i-te kolone, i naći odgovarajuću permutacionu matricu F_i. Proces se završava matricom

$$A_r = \begin{bmatrix} \Delta_{r,r} & X_{r,n-r} \\ \mathbb{O} & \mathbb{O} \end{bmatrix},$$

pri čemu se A_r dobija iz

$$Q^* = \prod_{i=1}^{r} \prod_{j=i+1}^{n} T_{ij}, \quad F = \prod_{k=1}^{r} F_k, \quad Q^*AF = A_r.$$

Odavde je $A = QA_rF^* = QR$. $\square$

Teorema 2.3 *Ako $A = QR$ predstavlja QR faktorizaciju matrice $A \in \mathbb{C}_r^{m\times n}$, tada je*

$$A^\dagger = R^\dagger Q^* = R^*(RR^*)^{-1}Q^*.$$

Metode za izračunavanje Moore-Penroseovog inverza, zasnovane na LU i QR faktorizaciji opisane su u [46, 58].

QR faktorizacija je u MATHEMATICA implementirana pomoću standardne funkcije *QRDecomposition*:

`QRDecomposition[m]` produkuje QR dekompoziciju numeričke matrice m. Rezultat je lista $\{q, r\}$, gde je q ortogonalna matrica, a r je gornja trougaona matrica. Polazna matrica m je jednaka sa

`Conjugate[Transpose[q]].r.`

Sada se izračunavanje Moore-Penroseovog inverza pomoću QR dekompozicije može implementirati sledećom funkcijom:

```
QRMP[a_]:=
    Block[{b=a,q,r},
      {q,r}=QRDecomposition[b];
      Return[Hermit[r].Inverse[r.Hermit[r]].q];
    ]
```

Primer 2.3 Data je matrica $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix}$.

In[1]:= a={{1, 2, 3}, {3, 2, 1.}}
In[2]:= {q,r}=QRDecomposition[a]
Out[2]={{{-0.316228, -0.948683}, {0.948683, -0.316228}},{{-3.16228, -2.52982, -1.89737}, {0, 1.26491, 2.52982}}}
In[3]:=Transpose[q].r
Out[3]= {{1., 2., 3.}, {3., 2., 1.}}
In[4]:= x=Transpose[r].Inverse[r.Transpose[r]].q
Out[5]= {{-0.166667, 0.333333}, {0.0833333, 0.0833333},{0.333333, -0.166667}}

Jednačine (1)–(4) se mogu proveriti:

$$\begin{aligned} axa &= \begin{bmatrix} 1. & 2. & 3. \\ 3. & 2. & 1. \end{bmatrix}, \quad xax = \begin{bmatrix} -0.166667 & 0.333333 \\ 0.0833333 & 0.0833333 \\ 0.333333 & -0.166667 \end{bmatrix} \\ ax &= \begin{bmatrix} 1. & -3.88578 * 10^{-16} \\ 2.77556 * 10^{-16} & 1. \end{bmatrix}, \\ xa &= \begin{bmatrix} 0.833333 & 0.333333 & -0.166667 \\ 0.333333 & 0.333333 & 0.333333 \\ -0.166667 & 0.333333 & 0.833333 \end{bmatrix} \end{aligned}$$

Penrose ([47], 408. str.) je predložio metod za direktno izračunavanje Moore-Penroseovog inverza pomoću singularno-vrednosne dekompozicije.

Teorema 2.4 *Ako je zadata singularno-vrednosna dekompozicija matrice A*

$$A = U^T \begin{bmatrix} \sum & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} V = U^T \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \mathbb{O} \\ & & \sigma_r & \\ & & \mathbb{O} & \mathbb{O} \end{bmatrix} V,$$

gde su U i V ortogonalne matrice, tada je

$$A^\dagger = V^T \begin{bmatrix} \sum^{-1} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} U = V^T \begin{bmatrix} \frac{1}{\sigma_1} & & & \\ & \ddots & & \mathbb{O} \\ & & \frac{1}{\sigma_r} & \\ & & \mathbb{O} & \mathbb{O} \end{bmatrix} U.$$

U ovim jednačinama su $\sigma_1 \leq \cdots \leq \sigma_r > 0$ singularne vrednosti matrice A, tj. sopstvene vrednosti matrice A^*A.

Singularno vrednosna dekompozicija se u MATHEMATICA može implementirati pomoću funkcije *SingularValues*:

Rezultat izraza `SingularValues[m]` je lista $\{u, w, v\}$, gde je w lista nenultih singularnih vrednosti za m, dok matrica m može biti napisana u obliku

`Transpose[u].DiagonalMatrix[w].v.`

Sada se izračunavanje Moore-Penroseovog inverza zasnovano na singularno vrednosnoj dekompoziciji može implementirati sledećom funkcijom:

```
SVDMP[a_]:=
    Block[{b=a,u,w,v},
       {u,w,v}=SingularValues[b];
       Return[Transpose[v].DiagonalMatrix[1/w].u];
    ]
```

Primer 2.4 Neka je $a = \begin{pmatrix} 3 & 1 & 4 & 9 \\ 1 & 2 & 3 & 4 \\ 0 & -2 & -2 & 0 \\ -1 & 0 & -1 & -4 \end{pmatrix}$.

In[1]:=a={{3,1,4,9},{1,2,3,4},{0,-2,-2,0},{-1,0,-1,-4}}

In[2]:= {u,w,v}=SingularValues[a]

Out[2]={{{-0.836633, -0.425092, 0.0921527, 0.33294},{0.206738, -0.480072, 0.793008, -0.312935},

{0.507251, -0.505464, -0.171211, 0.676675}},{12.3346, 3.26459, 0.447011},

{{-0.26494, -0.151697, -0.416637, -0.856276},{0.138786, -0.716605, -0.577819, 0.36516},

{0.759745, -0.360739, 0.399006, -0.365308}}}

In[3]:= Transpose[u].DiagonalMatrix[w].v

Out[3]={{3., 1., 4., 9.}, {1., 2., 3., 4.},{-1.47045*10^{-16}, -2., -2., 1.10995*10^{-15}},

{-1., 2.74832*10^{-16} , -1., -4.}}

In[4]:= x=Transpose[v].DiagonalMatrix[1/w].u

Out[4]={{0.888889, -0.87037, -0.259259, 1.12963},{-0.444444, 0.518519, -0.037037, -0.481481},

{0.444444, -0.351852, -0.296296, 0.648148},{-0.333333, 0.388889, 0.222222, -0.611111}}

Dobijena matrica x predstavlja Moore-Penroseov inverz matrice a:

$$\begin{aligned} a.x.a &= \begin{bmatrix} 3. & 1. & 4. & 9. \\ 1. & 2. & 3. & 4. \\ 3.33067 * 10^{-16} & -2. & -2. & 0. \\ -1. & -2.22045 * 10^{-16} & -1. & -4. \end{bmatrix}, \\ x.a.x &= \begin{bmatrix} 0.888889 & -0.87037 & -0.259259 & 1.12963 \\ -0.444444 & 0.518519 & -0.037037 & -0.481481 \\ 0.444444 & -0.351852 & -0.296296 & 0.648148 \\ -0.333333 & 0.388889 & 0.222222 & -0.611111 \end{bmatrix} \end{aligned}$$

$$ax = \begin{bmatrix} 1. & -5.55112*10^{-17} & -4.44089*10^{-16} & 8.88178*10^{-16} \\ -2.22045*10^{-16} & 0.666667 & -0.333333 & -0.333333 \\ 4.44089*10^{-16} & -0.333333 & 0.666667 & -0.333333 \\ -2.22045*10^{-16} & -0.333333 & -0.333333 & 0.666667 \end{bmatrix},$$

$$xa = \begin{bmatrix} 0.666667 & -0.333333 & 0.333333 & 8.88178*10^{-16} \\ -0.333333 & 0.666667 & 0.333333 & -2.22045*10^{-16} \\ 0.333333 & 0.333333 & 0.666667 & 4.44089*10^{-16} \\ 0. & -2.22045*10 & -16 & -2.22045*10^{-16} \end{bmatrix}$$

Sledeća metoda za izračunavanje Drazinovog pseudoinverza potiče od [9]. U njoj se može koristiti Jordanova kanonička forma.

Teorema 2.5 *Ako je $A \in \mathbb{C}^{n\times n}$ i $ind(A) = k > 0$, tada postoji nesingularna matrica T, takva da je*

$$A = T \begin{bmatrix} R & \mathbb{O} \\ \mathbb{O} & N \end{bmatrix} T^{-1},$$

gde je R nesingularna matrica, a N nilpotentna matrica indeksa k. Drazinov pseudoinverz matrice A je

$$A_d = T \begin{bmatrix} R^{-1} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} T^{-1}.$$

Napomenimo da se umesto matrice R može koristiti matrica dobijena od nesingularnih blokova Jordanove kanoničke forme za matricu A, dok se u ulozi matrice N može koristiti matrica izgrađena od nilpotentnih blokova Jordanove kanoničke forme za A.

Sledeći metod za izračunavanje Drazinovog pseudoinverza se zasniva na njegovoj polinomskoj reprezentaciji. Za njegovu primenu je neophodno poznavanje svih sopstvenih vrednosti polazne matrice i njihova višestrukost [57].

Teorema 2.6 *Neka je $A \in \mathbb{C}^{n\times n}$, i neka su λ_i, $i = 0, 1, \ldots, t$ različite sopstvene vrednosti matrice A sa odgovarajućim višestrukostima m_i. Neka je $\lambda_0 = 0$ i $m = m_1 + \ldots + m_t = n - m_0$. Ako je*

$$p(x) = x^{m_0} \left(\lambda_0 + \lambda_1 x + \ldots + \lambda_{m-1} x^{m-1}\right)$$

polinom čiji se koeficijenti određuju kao jedinstveno rešenje sistema linearnih *jednačina*

$$\begin{aligned}
\frac{1}{\lambda_i} &= p(\lambda_i),\\
-\frac{1}{\lambda_i^2} &= p'(\lambda_i),\\
&\vdots\\
\frac{(-1)^{m_i-1}(m_i-1)!}{\lambda_i^{m_i}} &= p^{(m_i-1)}(\lambda_i),
\end{aligned}$$

tada je $A_d = p(A)$.

2.1.3 Generalisani inverzi blokovskih matrica

Stav 2.2 *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* [86, 88]*:*

$$\begin{aligned}
&(T_1) \qquad RAG = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1\\
&(T_2) \qquad RAG = \begin{bmatrix} B & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_2\\
&(T_3) \qquad RAF = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_3\\
&(T_4) \qquad EAG = \begin{bmatrix} I_r & \mathbb{O} \\ K & \mathbb{O} \end{bmatrix} = N_4\\
&(T_5) \qquad UAG = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1\\
&(T_6) \qquad RAV = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1\\
&(T_7) \qquad UAV = \begin{bmatrix} B & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_2\\
&(T_8) \qquad UAF = \begin{bmatrix} B & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_5\\
&(T_9) \qquad EAV = \begin{bmatrix} B & \mathbb{O} \\ K & \mathbb{O} \end{bmatrix} = N_6\\
&(T_{10a}) \qquad EAF = \begin{bmatrix} A_{11} & A_{11}T \\ SA_{11} & SA_{11}T \end{bmatrix},
\end{aligned}$$

gde su S i T multiplikatori dobijeni iz uslova [46]

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

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

Transformacija sličnosti kvadratnih matrica [56].

$$(T_{11}) \quad RAR^{-1} = RAEE^{*}R^{-1} = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} E^{*}R^{-1} = \begin{bmatrix} T_1 & T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix}.$$

Regularna matrica R i permutaciona matrica E mogu se dobiti izvršavanjem transformacija koje su analogne transformacijama nad vrstama matrice A, nad jediničnom matricom I_m, dok se regularna matrica G i permutaciona matrica F mogu se dobiti sprovođenjem analognih transformacija koje su sprovedene nad kolonama matrice A, na jediničnoj matrici I_n.

Implementacija blokovske reprezentacije (T_3) sadržana je u sledećoj proceduri [42, 44, 74].

```
t3[a_]:=
  Block[{m,n,r,f,b=a,i=1,j,max=1,g,h,p1,p2},
   {m,n}=Dimensions[b];
   r=IdentityMatrix[m];    f=IdentityMatrix[n];
   While[max != 0,
           max=Abs[b[[i,i]]]; p1=i; p2=i;
           Do[Which[max<Abs[b[[g,h]]],
                      max=Abs[b[[g,h]]];p1=g;p2=h
                    ],   {g,i+1,m},{h,i+1,n}];
           If[max!=0,
                 r=ChRow[r,i,p1];f=ChColumn[f,i,p2];
                 b=ChRow[b,i,p1];b=ChColumn[b,i,p2];
                 r=MultRow[r,i,1/b[[i,i]]];
                 b=MultRow[b,i,1/b[[i,i]]];
                 Do[Which[j!=i,r=AddRow[r,j,i,-b[[j,i]]];
                             b=AddRow[b,j,i,-b[[j,i]]]
                          ], {j,m}];
                 i+=1
   ]      ];
   {r,f}
  ];MatrixQ[a]
```

Blokovska dekompozicija (T_1) može se generisati primenom transformacije (T_3) dva puta:

$$\begin{aligned} R_1 A F_1 &= \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_3, \\ R_2 N_3^T F_2 &= \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1. \end{aligned}$$

Tada, regularne matrice R, G mogu da se izračunaju sa:

$$N_1 = N_1^T = F_2^T N_3 R_2^T = F_2^T R_1 A F_1 R_2^T \Rightarrow R = F_2^T R_1, \quad G = F_1 R_2^T.$$

```
t1[a_]:=
  Block[{r1,f1,r2,f2,b=a},
          {r1,f1}=t3[b];
          {r2,f2}=t3[Transpose[r1.b.f1]];
          {Transpose[f2].r1,f1.Transpose[r2]}
  ];MatrixQ[a]
```

Blokovska dekompozicija (T_4) može se implementirati pomoću transformacije (T_3) na matrici A^T:

$$R_1 A^T F_1 = \begin{bmatrix} I_r & K_1 \\ \mathbb{O} & \mathbb{O} \end{bmatrix}.$$

Prema tome,

$$F_1^T A R_1^T = \begin{bmatrix} I_r & \mathbb{O} \\ K_1^T & \mathbb{O} \end{bmatrix},$$

što povlači $E = F_1^T$, $G = R_1^T$.

```
t4[a_]:=
  Block[{r1,f1,b=a},
          {r1,f1}=t3[Transpose[b]];
          {Transpose[f1],Transpose[r1]}
  ];MatrixQ[a]
```

Primer 2.5 Neka je

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 6 & 3 \\ -1 & 4 & 2 \end{bmatrix}.$$

Niz transformacija je sledeći:

$$\begin{bmatrix} 1 & 2 & 1 & & 1 & 0 & 0 & & 1 & 0 & 0 \\ 0 & 6 & 3 & & 0 & 1 & 0 & & 0 & 1 & 0 \\ -1 & 4 & 2 & & 0 & 0 & 1 & & 0 & 0 & 1 \end{bmatrix} \rightarrow$$

$$\left[\begin{array}{rrrcccccc} 6 & 0 & 3 & 0 & 1 & 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 4 & -1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \rightarrow$$

$$\left[\begin{array}{rrrcccccc} 1 & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 4 & -1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \rightarrow$$

$$\left[\begin{array}{rrrcccccc} 1 & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & -\frac{1}{3} & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -\frac{2}{3} & 1 & 0 & 0 & 1 \end{array}\right] \rightarrow$$

$$\left[\begin{array}{rrrcccccc} 1 & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & -\frac{1}{3} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Iz druge i treće matrice čitamo R i F:

$$R = \begin{bmatrix} 0 & \frac{1}{6} & 0 \\ 1 & -\frac{1}{3} & 0 \\ 1 & -1 & 1 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pozivom programa dobijamo:

In[2]:=t3[a]

Out[2]=$\{\{\{0, \frac{1}{6}, 0\}, \{1, -\frac{1}{3}, 0\}, \{1, -1, 1\}\}, \{\{0, 1, 0\}, \{1, 0, 0\}, \{0, 0, 1\}\}\}$

Metod za izračunavanje matrica S, T i A_{11}^{-1} dat je u [46]

$$\begin{bmatrix} A_{11} & A_{12} & I \\ A_{21} & A_{22} & \mathbb{O} \end{bmatrix} \rightarrow \begin{bmatrix} I_r & A_{11}^{-1}A_{12} & A_{11}^{-1} \\ \mathbb{O} & A_{22} - SA_{12} & -S \end{bmatrix} = \begin{bmatrix} I_r & T & A_{11}^{-1} \\ \mathbb{O} & \mathbb{O} & -S \end{bmatrix}.$$

Očigledno je $T = K$.

Primer 2.6 Za matricu $A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}$ može se dobiti

$$
\begin{aligned}
A_{11}^{-1} &= \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix}, \\
S &= \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}, \\
T &= \begin{bmatrix} -1 & -2 \\ -1 & -3 \end{bmatrix} [81]
\end{aligned}
$$

Iz blokovskih dekompozicija matrica sledi čitav niz reprezentacija za različite klase generalisanih inverza.

Prikaz poznatih blokovskih reprezentacija za različite klase generalisanih inverza, nalazi se u [86, 87, 88].

Teorema 2.7 [89] *Moore-Penroseov inverz matrice $A \in \mathbb{C}^{m\times n}$ jednak je*

$$A^\dagger = A^*(A^*AA^*)^{(1)}A^*.$$

Stav 2.3 [84] *Neka je $A \in \mathbb{C}_r^{m\times n}$ predstavljena u obliku $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, $A_{11} \in \mathbb{C}_r^{r\times r}$. Tada je*

$$A^\dagger = \begin{bmatrix} A_{11}, & A_{12} \end{bmatrix}^* T_{11}^* \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}^*,$$

gde je

$$T_{11} = \left[\begin{bmatrix} A_{11}, & A_{12} \end{bmatrix} A^* \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} \right]^{-1}.$$

Može se koristiti i Hermiteov algoritam, koji se zasniva na sledećoj formuli

$$A^\dagger = A^*(AA^*AA^*)^{(1,2)}A^* = \left((AA^*)^2\right)^{(1,2)}.$$

Pri tome, za izračunavanje inverza $((AA^*)^2)^{(1,2)}$ može se iskoristiti, na primer njegova blokovska reprezentacija iz Stavu 2.3, za $X = Y = \mathbb{O}$. Ako je $(AA^*)^2 = R^T I_r G$, tada je $((AA^*)^2)^{(1,2)} = G^T I_r R = (G^T)_r^{|r} R_{|r}$, gde $(G^T)^{|r}$ i $R_{|r}$ predstavljaju prvih r kolona matrice G^T i prvih r vrsta matrice R, respektivno.

Takođe, poznate su i blokovske reprezentacije grupnog i Drazinovog inverza.

Stav 2.4 [56] *Neka je* $A \in \mathbb{C}_r^{n\times n}$, *i* R *regularna matrica, tako da važi* $RAR^{-1} = \begin{bmatrix} T_1 & T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix}$. *Tada je*

$$A^{\#} = R^{-1} \begin{bmatrix} T_1^{-1} & T_1^{-2}T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix} R.$$

Ovaj rezultat je uopštio Hartwig, čime je razvio metod za izračunavanje Drazinovog inverza.

Stav 2.5 [36] *Ako je* $A \in \mathbb{C}_r^{n\times n}$, *i* R *nesingularna matrica, tako da je* $RAR^{-1} = \begin{bmatrix} U & V \\ \mathbb{O} & \mathbb{O} \end{bmatrix}$. *Tada je*

$$A_d = R^{-1} \begin{bmatrix} U_d & U_d^2 V_d \\ \mathbb{O} & \mathbb{O} \end{bmatrix} R.$$

Teorema 2.8 *Uočimo* $A \in \mathbb{C}_r^{m\times n}$ *i matrice* W_1, W_2 *kao i*

$$\begin{aligned} G^{-1}W_1 &= \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}, \\ W_2R^{-1} &= \begin{bmatrix} V_1, & V_2 \end{bmatrix}, \\ F^*W_1 &= \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}, \\ W_2E^* &= \begin{bmatrix} T_1, & T_2 \end{bmatrix}, \\ V^*W_1 &= \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \\ W_2U^* &= \begin{bmatrix} D_1, & D_2 \end{bmatrix}, \\ RW_1 &= \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \end{aligned}$$

gde U_1, S_1, C_1, Q_1 *označavaju prvih* r *vrsta odgovarajućih blokova, dok* V_1, T_1, D_1 *predsatavljaju prvih* r *kolona odgovarajućih blokova. Tada se može dobiti sledeća blokovska reprezentacija* $\{1,2\}$ *inverza za* A:

$$(G_{1,2}^1) \qquad A\{1,2\} = G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}(V_1U_1)^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R,$$

$$(G_{1,2}^2) \qquad A\{1,2\} = G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}(V_1BU_1)^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R,$$

$$(G_{1,2}^3) \qquad A\{1,2\} = F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}(S_1+KS_2)^{-1}V_1^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R,$$

$$(G_{1,2}^4) \qquad A\{1,2\} = G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}U_1^{-1}(T_1+T_2K)^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E,$$

$$(G_{1,2}^5) \qquad A\{1,2\} = G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}(D_1U_1)^{-1}\begin{bmatrix} D_1, & D_2 \end{bmatrix}U,$$

$$(G_{1,2}^6) \qquad A\{1,2\} = V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}(V_1C_1)^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R,$$

$$(G_{1,2}^7) \qquad A\{1,2\} = V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}(D_1BC_1)^{-1}\begin{bmatrix} D_1, & D_2 \end{bmatrix}U,$$

$$(G_{1,2}^8) \qquad A\{1,2\} = F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}(D_1(BS_1+KS_2))^{-1}\begin{bmatrix} D_1, & D_2 \end{bmatrix}U,$$

$$(G_{1,2}^9) \qquad A\{1,2\} = V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}((T_1B+T_2K)C_1)^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E,$$

$$(G_{1,2}^{10a}) \qquad A\{1,2\} = F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}((T_1+T_2S)A_{11}(S_1+TS_2))^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E,$$

$$(G_{1,2}^{10b}) \qquad A\{1,2\} = F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}(A_{11}S_1+A_{12}S_2)^{-1}A_{11}(T_1A_{11}+T_2A_{21})^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E,$$

$$(G_{1,2}^{11}) \qquad A\{1,2\} = R^{-1}\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}\left(V_1(T_1Q_1+T_2Q_2)^{-1}\right)^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R,$$

Dokaz. $(G_{1,2}^1)$ Prema (T_1) dobija se sledeća full-rang faktorizacija za A:

$$P = R^{-1}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}, \quad Q = \begin{bmatrix} I_r & \mathbb{O} \end{bmatrix}G^{-1}.$$

Sada je,

$$\begin{aligned} QW_1 &= \begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix}G^{-1}G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = U_1, \\ W_2P &= \begin{bmatrix} V_1, & V_2 \end{bmatrix}RR^{-1}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} = V_1. \end{aligned}$$

Odatle je:

$$A\{1,2\} = W_1(QW_1)^{-1}(W_2P)^{-1}W_2 = G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}(V_1U_1)^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R.$$

Matrice W_1 i W_2 ispunjavaju rank$(QW_1) = $ rank$(W_2P) = r$, tako da su U_1 i V_1 invertibilne matrice. Time je obezbeđena egzistencija blokovske reprezentacije $(G^1_{1,2})$. Egzistencija ostalih blokovskih reprezentacija $(G^i_{1,2}), \quad i \in \{2, \ldots, 11\}$ može da se pokaže analogno.

$(G^5_{1,2})$ Blokovska dekompozicija (T_5) implicira

$$P = U^* \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}, \qquad Q = \begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix} G^{-1}.$$

Ovaj deo dokaza se može kompletirati koristeći

$$\begin{aligned} W_1 = V \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}, \qquad QW_1 &= \begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix} G^{-1} G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = U_1, \\ W_2 = \begin{bmatrix} D_1, & D_2 \end{bmatrix} U, \qquad W_2P &= \begin{bmatrix} D_1, & D_2 \end{bmatrix} U U^* \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} = D_1. \end{aligned}$$

$(G^7_{1,2})$ Lako se proverava da iz (T_7) sledi

$$P = U^* \begin{bmatrix} B \\ \mathbb{O} \end{bmatrix}, \qquad Q = \begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix} V^*.$$

Prema tome,

$$QW_1 = D_1 B, \qquad W_2 P = C_1,$$

što proizvodi,

$$A\{1,2\} = V \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (D_1 B C_1)^{-1} \begin{bmatrix} D_1, & D_2 \end{bmatrix} U.$$

$(G^{10a}_{1,2})$ Iz (T_{10a}) dobija se sledeća full-rang faktorizacija za A:

$$P = E^* \begin{bmatrix} I_r \\ S \end{bmatrix} A_{11}, \quad Q = \begin{bmatrix} I_r, & T \end{bmatrix} F^*.$$

Prema tome,

$$\begin{aligned} QW_1 &= \begin{bmatrix} I_r, & T \end{bmatrix} F^* F \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = S_1 + T S_2, \\ W_2 P &= \begin{bmatrix} T_1, & T_2 \end{bmatrix} E E^* \begin{bmatrix} I_r \\ S \end{bmatrix} A_{11} = (T_1 + T_2 S) A_{11}. \qquad \square \end{aligned}$$

Sledi opis implementacije blokovskih reprezentacija $(G^1_{1,2})$, $(G^3_{1,2})$ i $(G^4_{1,2})$ za izračunavanje $\{1,2\}$ inverza, koje su date u poslednjoj teoremi. Biće iskorišćene funkcije $t1$, $t3$ i $t4$ kojima su implementirane blokovske dekompozicije (T_1), (T_3) i (T_4), respektivno [42, 30, 74].

```
Reflexive1[b_,r1_,r2_]:=
Block[{a=b,w1=r1,w2=r2, ran,r,g,u1,u2,v1,v2,pom1,pom2},
          {r,g}=t1[a]; ran=rank[a];
          pom1=Inverse[g].w1; pom2=w2.Inverse[r];
          u1=Take[pom1,ran];
          u2=Drop[pom1,ran];
          v1=Transpose[Take[Transpose[pom2],ran]];
          v2=Transpose[Drop[Transpose[pom2],ran]];
          Return[g.pom1.Inverse[v1.u1].pom2.r];
]

Reflexive3[b_,r1_,r2_]:=
Block[{a=b,w1=r1,w2=r2, ran,r,g,s1,s2,k,v1,v2,pom1,pom2,pom3},
          {r,f}=t3[a]; ran=rank[a];
          pom1=Hermit[f].w1; pom2=w2.Inverse[r];
          s1=Take[pom1,ran];
          s2=Drop[pom1,ran];
          v1=Transpose[Take[Transpose[pom2],ran]];
          v2=Transpose[Drop[Transpose[pom2],ran]];
          pom3=Take[r.a.f,ran];
          k=Transpose[Drop[Transpose[pom3],ran]];
          Return[f.pom1.Inverse[s1+k.s2].Inverse[v1].pom2.r];
]

Reflexive4[b_,r1_,r2_]:=
Block[{a=b,w1=r1,w2=r2, ran,r,g,u1,u2,k,t1,t2,pom1,pom2,pom3},
          {e,g}=t4[a]; ran=rank[a];
          pom1=Inverse[g].w1; pom2=w2.Hermit[e];
          u1=Take[pom1,ran];
          u2=Drop[pom1,ran];
          t1=Transpose[Take[Transpose[pom2],ran]];
          t2=Transpose[Drop[Transpose[pom2],ran]];
          pom3=drop[e.a.g,ran];
          k=Transpose[Take[Transpose[pom3],ran]];
          Return[g.pom1.Inverse[u1].Inverse[t1+t2.k].pom2.e];
]
```

Primer 2.7 Koristeći

$$A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}, \quad W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 4 & 2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & -1 \\ 0 & -1 & 0 & 0 & -2 & -1 \end{bmatrix}$$

izraz `x=Reflexive1[a,w1,w2]` dobija se vrednost

$$X = \begin{bmatrix} \frac{2}{17} & \frac{20}{51} & \frac{2}{17} & \frac{2}{51} & \frac{40}{51} & \frac{16}{51} \\ -\frac{1}{51} & -\frac{61}{153} & -\frac{1}{51} & -\frac{1}{153} & -\frac{122}{153} & -\frac{59}{153} \\ \frac{1}{17} & \frac{10}{51} & \frac{1}{17} & \frac{1}{51} & \frac{20}{51} & \frac{8}{51} \\ \frac{10}{51} & -\frac{2}{153} & \frac{10}{51} & \frac{10}{153} & -\frac{4}{153} & -\frac{22}{153} \end{bmatrix}$$

Provera:

In[1]:= a.x.a==a

Out[1]= True

In[2]:= x.a.x==x

Out[2]= True

In[3]:= a.x==Transpose[a.x]

Out[3]= False

In[4]:= x.a==Transpose[x.a]

Out[4]= False

Vrednost izraza `x=Reflexive3[a,w1,w2]` i `Reflexive4[a,w1,w2]` je ista matrica.

Glavne prednosti uvedenih blokovskih reprezentacija jesu njihova jednostavna derivacija i redukcija u blokovsku reprezentaciju $\{1,2,3\}$, $\{1,2,4\}$ inverza, Moore-Penroseovog inverza, težinskog Moore-Penroseovog inverza kao i grupnog inverza, bez rešavanja odgovarajućih matričnih jednačina.

Teorema 2.9 *Neka je $A \in \mathbb{C}_r^{m\times n}$ i neka su date matrice W_1 i W_2. Tada se može dobiti sledeće blokovske reprezentacije $\{1,2,3\}$ i $\{1,2,4\}$ generalisanih inverza za A, pod uslovom da $(G^i_{1,2,3})$ i $(G^i_{1,2,4})$ odgovaraju (T_i), $i \in \{1,\ldots 11\}$:*

$$
\begin{array}{lll}
(G^1_{1,2,3}) \quad A\{1,2,3\} & = & G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} U_1^{-1}\left((RR^*)^{-1}{}^{|r}_{|r}\right)^{-1}\begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix}(R^*)^{-1} \\
& = & G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}\left((RR^*)^{-1}{}^{|r}_{|r}U_1\right)^{-1}(R^*)^{-1}{}_{|r}, \\
(G^1_{1,2,4}) \quad A\{1,2,4\} & = & (G^*)^{-1}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}\left((G^*G)^{-1}{}^{|r}_{|r}\right)^{-1}V_1^{-1}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix}R \\
& = & (G^*)^{-1|r}\left(V_1(G^*G)^{-1}{}^{|r}_{|r}\right)^{-1}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix}R, \\
(G^2_{1,2,3}) \quad A\{1,2,3\} & = & G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} U_1^{-1}\left(B^*(RR^*)^{-1}{}^{|r}_{|r}B\right)^{-1}\begin{bmatrix} B^*, & \mathbb{O} \end{bmatrix}(R^*)^{-1} \\
& = & G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}\left((RR^*)^{-1}{}^{|r}_{|r}BU_1\right)^{-1}(R^*)^{-1}{}_{|r}, \\
(G^2_{1,2,4}) \quad A\{1,2,4\} & = & (G^*)^{-1}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}\left((G^*G)^{-1}{}^{|r}_{|r}\right)^{-1}(V_1B)^{-1}\begin{bmatrix} V_1, & B \end{bmatrix}R \\
& = & (G^*)^{-1|r}\left(V_1B(G^*G)^{-1}{}^{|r}_{|r}\right)^{-1}\begin{bmatrix} V_1, & B \end{bmatrix}R, \\
(G^3_{1,2,3}) \quad A\{1,2,3\} & = & F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}(S_1+KS_2)^{-1}\left((RR^*)^{-1}{}^{|r}_{|r}\right)^{-1}\begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix}(R^*)^{-1} \\
& = & F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}\left((RR^*)^{-1}{}^{|r}_{|r}(S_1+KS_2)\right)^{-1}(R^*)^{-1}{}_{|r}, \\
(G^3_{1,2,4}) \quad A\{1,2,4\} & = & F\begin{bmatrix} I_r \\ K^* \end{bmatrix}(I_r+KK^*)^{-1}V_1^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R \\
& = & \left(F^{|r}+F^{n-r|}K^*\right)(V_1(I_r+KK^*))^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R, \\
(G^4_{1,2,3}) \quad A\{1,2,3\} & = & G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}U_1^{-1}(I_r+KK^*)^{-1}\begin{bmatrix} I_r, & K^* \end{bmatrix}E \\
& = & G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}((I_r+KK^*)U_1)^{-1}\left(E_{|r}+K^*E_{n-r|}\right), \\
(G^4_{1,2,4}) \quad A\{1,2,4\} & = & (G^*)^{-1}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}\left((G^*G)^{-1}{}^{|r}_{|r}\right)^{-1}(T_1+T_2K)^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E \\
& = & (G^*)^{-1|r}\left((T_1+T_2K)(G^*G)^{-1}{}^{|r}_{|r}\right)^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E, \\
(G^5_{1,2,3}) \quad A\{1,2,3\} & = & G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}U_1^{-1}\begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix}U = G\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}U_1^{-1}U_{|r}, \\
(G^5_{1,2,4}) \quad A\{1,2,4\} & = & (G^*)^{-1}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}\left((G^*G)^{-1}{}^{|r}_{|r}\right)^{-1}D_1^{-1}\begin{bmatrix} D_1, & D_2 \end{bmatrix}U \\
& = & (G^*)^{-1|r}\left(D_1(G^*G)^{-1}{}^{|r}_{|r}\right)^{-1}\begin{bmatrix} D_1, & D_2 \end{bmatrix}U, \\
(G^6_{1,2,3}) \quad A\{1,2,3\} & = & V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}C_1^{-1}\left((RR^*)^{-1}{}^{|r}_{|r}\right)^{-1}\begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix}(R^*)^{-1} \\
& = & V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}\left((RR^*)^{-1}{}^{|r}_{|r}C_1\right)^{-1}(R^*)^{-1}{}_{|r}, \\
(G^6_{1,2,4}) \quad A\{1,2,4\} & = & V\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}V_1^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R = V^{|r}V_1^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R, \\
(G^7_{1,2,3}) \quad A\{1,2,3\} & = & V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}C_1^{-1}(B^*B)^{-1}\begin{bmatrix} B^*, & \mathbb{O} \end{bmatrix}U = V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}(BC_1)^{-1}U_{|r},
\end{array}
$$

$$
\begin{aligned}
(G^{7}_{1,2,4})\quad A\{1,2,4\} &= V\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}(D_1B)^{-1}\begin{bmatrix} D_1, & D_2 \end{bmatrix}U = V^{|r}(D_1B)^{-1}\begin{bmatrix} D_1, & D_2 \end{bmatrix}U,\\
(G^{8}_{1,2,3})\quad A\{1,2,3\} &= F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}(BS_1+KS_2)^{-1}\begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix}U\\
&= F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}(BS_1+KS_2)^{-1}U_{|r},\\
(G^{8}_{1,2,4})\quad A\{1,2,4\} &= F\begin{bmatrix} B^* \\ K^* \end{bmatrix}(D_1(BB^*+KK^*))^{-1}\begin{bmatrix} D_1, & D_2 \end{bmatrix}U,\\
(G^{9}_{1,2,3})\quad A\{1,2,3\} &= V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}((B^*B+K^*K)C_1)^{-1}\begin{bmatrix} B^*, & K^* \end{bmatrix}E,\\
(G^{9}_{1,2,4})\quad A\{1,2,4\} &= V\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix}(T_1B+T_2K)^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E\\
&= V^{|r}(T_1B+T_2K)^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E,\\
(G^{10a}_{1,2,3})\quad A\{1,2,3\} &= F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}(S_1+TS_2)^{-1}A_{11}^{-1}(I_r+S^*S)^{-1}\begin{bmatrix} I_r, & S^* \end{bmatrix}E\\
&= F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}((I_r+S^*S)A_{11}(S_1+TS_2))^{-1}\left(E_{|r}+S^*E_{n-r|}\right),\\
(G^{10a}_{1,2,4})\quad A\{1,2,4\} &= F\begin{bmatrix} I_r \\ T^* \end{bmatrix}(I_r+TT^*)^{-1}A_{11}^{-1}(T_1+T_2S)^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E\\
&= \left(F^{|r}+F^{n-r|}T^*\right)((T_1+T_2S)A_{11}(I_r+TT^*))^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E,\\
(G^{10b}_{1,2,3})\quad A\{1,2,3\} &= F\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}(A_{11}S_1+A_{12}S_2)^{-1}A_{11}(A_{11}^*A_{11}+A_{21}^*A_{21})^{-1}\begin{bmatrix} A_{11}^*, & A_{21}^* \end{bmatrix}E,\\
(G^{10b}_{1,2,4})\quad A\{1,2,4\} &= F\begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix}(A_{11}A_{11}^*+A_{12}A_{12}^*)^{-1}A_{11}(T_1A_{11}+T_2A_{21})^{-1}\begin{bmatrix} T_1, & T_2 \end{bmatrix}E,\\
(G^{11}_{1,2,3})\quad A\{1,2,3\} &= R^{-1}\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}(Q_1+T_1^{-1}T_2Q_2)^{-1}\left(T_1^*(RR^*)^{-1}{}^{|r}_{|r}T_1\right)^{-1}\begin{bmatrix} T_1^*, & \mathbb{O} \end{bmatrix}(R^{-1})^*\\
&= R^{-1}\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}\left(T_1^*(RR^*)^{-1}{}^{|r}_{|r}(T_1Q_1+T_2Q_2)\right)^{-1}(R^{-1|r}T_1)^*,\\
(G^{11}_{1,2,4})\quad A\{1,2,4\} &= R^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\left(V_1\begin{bmatrix} T_1, & T_2 \end{bmatrix}RR^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\right)^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix}R.
\end{aligned}
$$

Dokaz. Ove reprezentacije mogu da se razviju pomoću zamena $W_2 = P^*$ za $\{1,2,3\}$ inverze, i $W_1 = Q^*$ za $\{1,2,4\}$ inverza u odgovarajućim reprezentacijama $(G^{i}_{1,2})$. Na primer, $(G^{1}_{1,2,3})$ i $(G^{1}_{1,2,4})$ slede iz

$$
\begin{aligned}
QW_1 &= U_1, \quad W_2P = V_1,\\
QQ^* &= \begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix}(G^*G)^{-1}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} = (G^*G)^{-1}{}^{|r}_{|r},\\
P^*P &= \left[\begin{matrix} I_r, & \mathbb{O} \end{matrix}\right)(RR^*)^{-1}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} = (RR^*)^{-1}{}^{|r}_{|r}.
\end{aligned}
$$

U ostatku dokaza posmatraju se jedino sledeće transformacije:

$$
\begin{aligned}
(G^{7}_{1,2,3})\quad A\{1,2,3\} &= V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} C_1^{-1}(B^*B)^{-1}\begin{bmatrix} B^*, & \mathbb{O} \end{bmatrix} U \\
&= V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}(BC_1)^{-1}\begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix} U = V\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}(BC_1)^{-1}U_{|r}. \\
(G^{11}_{1,2,4})\quad A\{1,2,4\} &= R^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\left(\begin{bmatrix} I_r, & T_1^{-1}T_2 \end{bmatrix} RR^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\right)^{-1}(V_1T_1)^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix} R \\
&= R^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\left(V_1\begin{bmatrix} T_1, & T_2 \end{bmatrix} RR^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\right)^{-1}\begin{bmatrix} V_1, & V_2 \end{bmatrix} R. \qquad \square
\end{aligned}
$$

Napomena 2.1 *Reprezentacije* $(G^{i}_{1,2,3})$ *i* $(G^{i}_{1,2,4})$ *ne zahtevaju izračunavanje Moore-Penroseovog inverza nekih blokova, što je neophodno u* [86, 88].

```
Reflexive31[b_,r1_,r2_]:=
Block[a=b,w1=r1,w2=r2, ran,r,g,u1,u2,rz,pom1,pom2,
   {r,g}=t1[a]; ran=rank[a];
   pom1=Inverse[g].w1; pom2=w2.Inverse[r];
   u1=Take[pom1,ran];
   u2=Drop[pom1,ran];
     rz=Inverse[r.Hermit[r]];
   rz=Take[rz,ran];
     rz=Transpose[Take[Transpose[rz],ran]];
   Return[g.pom1.Inverse[rz.u1].Take[Inverse[Hermit[r]],ran]];
]

Reflexive41[b_,r1_,r2_]:=
Block[a=b,w1=r1,w2=r2, ran,r,g,v1,v2,gz,gz1,pom2,
   {r,g}=t1[a]; ran=rank[a];
   pom2=w2.Inverse[r];
   v1=Transpose[Take[Transpose[pom2],ran]];
   v2=Transpose[Drop[Transpose[pom2],ran]];
   gz=Inverse[Hermit[g].g];
     gz=Take[gz,ran];
     gz=Transpose[Take[Transpose[gz],ran]];
     gz1=Transpose[Take[Transpose[Inverse[Hermit[g]]],ran]];
   Return[gz1.Inverse[v1.gz].pom2.r];
]

Reflexive33[b_,r1_,r2_]:=
Block[{a=b,w1=r1,w2=r2, ran,r,f,s1,s2,k,v1,v2,pom1,pom3},
   {r,f}=t3[a]; ran=rank[a];
```

```
   pom1=Hermit[f].w1;
   s1=Take[pom1,ran];
   s2=Drop[pom1,ran];
   pom3=Take[r.a.f,ran];
   k=Transpose[Drop[Transpose[pom3],ran]];
     rz=Inverse[r.Hermit[r]];
     rz=Take[rz,ran];
   rz=Transpose[Take[Transpose[rz],ran]];
   Return[f.pom1.Inverse[rz.(s1+k.s2)]
        .Take[Inverse[Hermit[r]],ran]];
]

Reflexive43[b_,r1_,r2_]:=
Block[{a=b,w1=r1,w2=r2, ran,r,f,v1,v2,k,pom2,pom3},
   {r,f}=t3[a]; ran=rank[a];
   pom2=w2.Inverse[r];
   v1=Transpose[Take[Transpose[pom2],ran]];
   v2=Transpose[Drop[Transpose[pom2],ran]];
   pom3=Take[r.a.f,ran];
   k=Transpose[Drop[Transpose[pom3],ran]];
   Return[f.Hermit[pom3].Inverse[IdentityMatrix[ran]+k.Hermit[k]].
                  Inverse[v1].pom2.r];
]

Reflexive34[b_,r1_,r2_]:=
Block[{a=b,w1=r1,w2=r2, ran,e,g,u1,u2,k,pom1,pom3},
   {e,g}=t4[a]; ran=rank[a];
   pom1=Inverse[g].w1;
   u1=Take[pom1,ran];
   u2=Drop[pom1,ran];
   pom3=Drop[e.a.g,ran];
   k=Transpose[Take[Transpose[pom3],ran]];
   pom3=Conjugate[Take[Transpose[e.a.g],ran]];
   Return[g.pom1.Inverse[(IdentityMatrix[ran]
         +k.Hermit[k]).u1].pom3.e];
]

Reflexive44[b_,r1_,r2_]:=
Block[{a=b,w1=r1,w2=r2, ran,e,g,u1,u2,k,gz,t1,t2,pom2,pom3},
   {e,g}=t4[a]; ran=rank[a];
   pom2=w2.Hermit[e];
   t1=Transpose[Take[Transpose[pom2],ran]];
```

```
        t2=Transpose[Drop[Transpose[pom2],ran]];
        pom3=Drop[e.a.g,ran];
        k=Transpose[Take[Transpose[pom3],ran]];
        gz=Inverse[Hermit[g].g];
        gz=Take[gz,ran];
        gz=Transpose[Take[Transpose[gz],ran]];
        gz1=Transpose[Take[Transpose[Inverse[Hermit[g]]],ran]];
        Return[gz1.Inverse[(t1+t2.k).gz].pom2.e];
    ]
```

Primer 2.8 Koristeći

$$A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}, \quad W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 4 & 2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & -1 \\ 0 & -1 & 0 & 0 & -2 & -1 \end{bmatrix}$$

izraz `x=Reflexive1[a,w1,w2]` dobija vrednost

$$X = \begin{bmatrix} -\frac{2}{17} & -\frac{3}{17} & \frac{1}{17} & -\frac{1}{17} & \frac{3}{17} & \frac{2}{17} \\ \frac{19}{102} & \frac{10}{51} & -\frac{1}{102} & \frac{1}{102} & -\frac{10}{51} & -\frac{19}{102} \\ -\frac{1}{17} & -\frac{3}{34} & \frac{1}{34} & -\frac{1}{34} & \frac{3}{34} & \frac{1}{17} \\ \frac{7}{51} & \frac{2}{51} & \frac{5}{51} & -\frac{5}{51} & -\frac{2}{51} & -\frac{7}{51} \end{bmatrix}$$

Provera:

In[1]:= a.x.a==a
Out[1]= True
In[2]:= x.a.x==x
Out[2]= True
In[3]:= a.x==Transpose[a.x]
Out[3]= True
In[4]:= x.a==Transpose[x.a]
Out[4]= False

U slučaju $W_1 = Q^*$, $W_2 = P^*$ dobija se ovakva reprezentacija Moore-Penroseovog inverza.

Posledica 2.1 *Moore-Penroseov inverza date matrice* $A \in \mathbb{C}_r^{m\times n}$ *se može predstaviti kako sledi, pri čemu reprezentacije* $(A_\dagger^i)$ *odgovaraju blokovskim dekompozicijama* (T_i), $i \in \{1, \ldots, 9, 10a, 10b, 11\}$*:*

$$
\begin{aligned}
(A_{\dagger}^{1})\quad A^{\dagger} &= \left(G^{-1}{}_{|r}\right)^{*}\left(\left(R^{-1^{|r}}\right)^{*}A\left(G^{-1}{}_{|r}\right)^{*}\right)^{-1}\left(R^{-1^{|r}}\right)^{*}\\
&= \left(G^{-1}{}_{|r}\right)^{*}\left((RR^{*})^{-1}{}^{|r}_{|r}(G^{*}G)^{-1}{}^{|r}_{|r}\right)^{-1}\left(R^{-1^{|r}}\right)^{*},\\
(A_{\dagger}^{2})\quad A^{\dagger} &= \left(G^{-1}{}_{|r}\right)^{*}\left(\left(R^{-1^{|r}}B\right)^{*}A\left(G^{-1}{}_{|r}\right)^{*}\right)^{-1}\left(R^{-1^{|r}}B\right)^{*}\\
&= \left(G^{-1}{}_{|r}\right)^{*}\left(B^{*}(RR^{*})^{-1}{}^{|r}_{|r}B(G^{*}G)^{-1}{}^{|r}_{|r}\right)^{-1}B^{*}\left(R^{-1^{|r}}\right)^{*},\\
(A_{\dagger}^{3})\quad A^{\dagger} &= F\begin{bmatrix} I_r\\ K^{*}\end{bmatrix}\left(\left(R^{-1^{|r}}\right)^{*}AF\begin{bmatrix} I_r\\ K^{*}\end{bmatrix}\right)^{-1}\left(R^{-1^{|r}}\right)^{*}\\
&= \left(F^{|r}+F^{n-r|}K\right)\left((RR^{*})^{-1}{}^{|r}_{|r}(I_r+KK^{*})\right)^{-1}\left(R^{-1^{|r}}\right)^{*},\\
(A_{\dagger}^{4})\quad A^{\dagger} &= \left(G^{-1}{}_{|r}\right)^{*}\left(\begin{bmatrix} I_r, & K^{*}\end{bmatrix}EA\left(G^{-1}{}_{|r}\right)^{*}\right)^{-1}\begin{bmatrix} I_r, & K^{*}\end{bmatrix}E\\
&= \left(G^{-1}{}_{|r}\right)^{*}\left((I_r+K^{*}K)(G^{*}G)^{-1}{}^{|r}_{|r}\right)^{-1}\left(E_{|r}+K^{*}E_{n-r|}\right),\\
(A_{\dagger}^{5})\quad A^{\dagger} &= \left(G^{-1}{}_{|r}\right)^{*}\left(U_{|r}A\left(G^{-1}{}_{|r}\right)^{*}\right)^{-1}U_{|r}=\left(G^{-1}{}_{|r}\right)^{*}\left((G^{*}G)^{-1}{}^{|r}_{|r}\right)^{-1}U_{|r},\\
(A_{\dagger}^{6})\quad A^{\dagger} &= V^{|r}\left(\left(R^{-1^{|r}}\right)^{*}AV^{|r}\right)^{-1}\left(R^{-1^{|r}}\right)^{*}=V^{|r}\left((RR^{*})^{-1}{}^{|r}_{|r}\right)^{-1}\left(R^{-1^{|r}}\right)^{*},\\
(A_{\dagger}^{7})\quad A^{\dagger} &= V^{|r}\left(B^{*}U_{|r}AV^{|r}\right)^{-1}B^{*}U_{|r}=V^{|r}B^{-1}U_{|r},\\
(A_{\dagger}^{8})\quad A^{\dagger} &= F\begin{bmatrix} B^{*}\\ K^{*}\end{bmatrix}\left(U_{|r}AF\begin{bmatrix} B^{*}\\ K^{*}\end{bmatrix}\right)^{-1}U_{|r}=F\begin{bmatrix} B^{*}\\ K^{*}\end{bmatrix}(BB^{*}+KK^{*})^{-1}U_{|r},\\
(A_{\dagger}^{9})\quad A^{\dagger} &= V^{|r}\left(\begin{bmatrix} B^{*}, & K^{*}\end{bmatrix}EAV^{|r}\right)^{-1}\begin{bmatrix} B^{*}, & K^{*}\end{bmatrix}E\\
&= V^{|r}(B^{*}B+K^{*}K)^{-1}\begin{bmatrix} B^{*}, & K^{*}\end{bmatrix}E,\\
(A_{\dagger}^{10a})\quad A^{\dagger} &= F\begin{bmatrix} I_r\\ T^{*}\end{bmatrix}\left(A_{11}^{*}\begin{bmatrix} I_r, & S^{*}\end{bmatrix}EAF\begin{bmatrix} I_r\\ T^{*}\end{bmatrix}\right)^{-1}A_{11}^{*}\begin{bmatrix} I_r, & S^{*}\end{bmatrix}E\\
&= \left(F^{|r}+F^{n-r|}T^{*}\right)\left((I_r+S^{*}S)A_{11}(I_r+TT^{*})\right)^{-1}\left(E_{|r}+S^{*}E_{n-r|}\right),\\
(A_{\dagger}^{10b})\quad A^{\dagger} &= F\begin{bmatrix} A_{11}^{*}\\ A_{12}^{*}\end{bmatrix}\left((A_{11}^{*})^{-1}\begin{bmatrix} A_{11}^{*}, & A_{21}^{*}\end{bmatrix}EAF\begin{bmatrix} A_{11}^{*}\\ A_{12}^{*}\end{bmatrix}\right)^{-1}(A_{11}^{*})^{-1}\begin{bmatrix} A_{11}^{*}, & A_{21}^{*}\end{bmatrix}E\\
&= F\begin{bmatrix} A_{11}^{*}\\ A_{12}^{*}\end{bmatrix}(A_{11}A_{11}^{*}+A_{12}A_{12}^{*})^{-1}A_{11}(A_{11}^{*}A_{11}+A_{21}^{*}A_{21})^{-1}\begin{bmatrix} A_{11}^{*}, & A_{21}^{*}\end{bmatrix}E,
\end{aligned}
$$

$$
\begin{aligned}
(A_{\dagger}^{11})\quad A^{\dagger} &= R^{*}\begin{bmatrix} I_r\\ \left(T_1^{-1}T_2\right)^{*}\end{bmatrix}\left(\left(R^{-1^{|r}}T_1\right)^{*}AR^{*}\begin{bmatrix} I_r\\ \left(T_1^{-1}T_2\right)^{*}\end{bmatrix}\right)^{-1}\left(R^{-1^{|r}}T_1\right)^{*}\\
&= R^{*}\begin{bmatrix} I_r\\ \left(T_1^{-1}T_2\right)^{*}\end{bmatrix}\left(T_1^{*}(RR^{*})^{-1}{}^{|r}_{|r}\begin{bmatrix} T_1, & T_2\end{bmatrix}RR^{*}\begin{bmatrix} I_r\\ \left(T_1^{-1}T_2\right)^{*}\end{bmatrix}\right)^{-1}\left(R^{-1^{|r}}T_1\right)^{*}.
\end{aligned}
$$

Implementacione procedure za izračunavanje Moore-Penroseovog inverza, bazirane na reprezentacijama $(G_{\dagger}^{1})$, $(G_{\dagger}^{3})$ i $(G_{\dagger}^{4})$ jesu:

```
MPt1[a_]:=
  Block[{b=a,r,g,ran,pt,qt},
      {r,g}=t1[b];  ran=rank[b];
      qt=Hermit[Take[Inverse[g],ran]];
      pt=Conjugate[Take[Transpose[Inverse[r]],ran]];
      qt.Inverse[pt.b.qt].pt
  ];MatrixQ[a]
MPt3[a_]:=
  Block[{b=a,r,f,ran,pt,qt},
      {r,f}=t3[b];  ran=rank[b];
      qt=f.Hermit[Take[r.a.f,ran]];
      pt=Conjugate[Take[Transpose[Inverse[r]],ran]];
      qt.Inverse[pt.b.qt].pt
  ];MatrixQ[a]
MPt4[a_]:=
  Block[{b=a,e,g,ran,pt,qt},
      {e,g}=t4[b];  ran=rank[b];
      qt=Hermit[Take[Inverse[g],ran]];
      pt=Conjugate[Take[Transpose[e.b.g],ran]].e;
      qt.Inverse[pt.b.qt].pt
  ];MatrixQ[a]
```

Teorema 2.10 *Težinski Moore-Penroseov inverz $A^{\dagger}_{M\circ,\circ N}$ za $A \in \mathbb{C}_r^{m\times n}$ poseduje sledeće blokovske reprezencije, gde reprezentacije (Z_i) slede iz blokovskih dekompozicija (T_i), $i \in \{1,\dots,9,10a,10b,11\}$:*

$(Z_1)\quad N\left(G^{-1}{}_{|r}\right)^*\left(\left(R^{-1^{|r}}\right)^* MAN\left(G^{-1}{}_{|r}\right)^*\right)^{-1}\left(R^{-1^{|r}}\right)^* M,$

$(Z_2)\quad N\left(G^{-1}{}_{|r}\right)^*\left(\left(R^{-1^{|r}}B\right)^* MAN\left(G^{-1}{}_{|r}\right)^*\right)^{-1}\left(R^{-1^{|r}}B\right)^* M,$

$(Z_3)\quad NF\begin{bmatrix} I_r \\ K^* \end{bmatrix}\left(\left(R^{-1^{|r}}\right)^* MANF\begin{bmatrix} I_r \\ K^* \end{bmatrix}\right)^{-1}\left(R^{-1^{|r}}\right)^* M,$

$(Z_4)\quad N\left(G^{-1}{}_{|r}\right)^*\left(\begin{bmatrix} I_r, & K^* \end{bmatrix} EMAN\left(G^{-1}{}_{|r}\right)^*\right)^{-1}\begin{bmatrix} I_r, & K^* \end{bmatrix} EM,$

$(Z_5)\quad N\left(G^{-1}{}_{|r}\right)^*\left(U_{|r}MAN\left(G^{-1}{}_{|r}\right)^*\right)^{-1} U_{|r}M,$

$(Z_6)\quad NV^{|r}\left(\left(R^{-1^{|r}}\right)^* MANV^{|r}\right)^{-1}\left(R^{-1^{|r}}\right)^* M,$

$(Z_7)\quad NV^{|r}\left(B^*U_{|r}MANV^{|r}\right)^{-1} B^*U_{|r}M,$

$(Z_8)\quad NF\begin{bmatrix} B^* \\ K^* \end{bmatrix}\left(U_{|r}MANF\begin{bmatrix} B^* \\ K^* \end{bmatrix}\right)^{-1} U_{|r}M,$

$(Z_9)\quad NV^{|r}\left(\begin{bmatrix} B^*, & K^* \end{bmatrix} EMANV^{|r}\right)^{-1}\begin{bmatrix} B^*, & K^* \end{bmatrix} EM,$

(Z_{10a}) $NF\begin{bmatrix} I_r \\ T^* \end{bmatrix}\left(A_{11}^*[\ I_r, \quad S^*\ ]\,EMANF\begin{bmatrix} I_r \\ T^* \end{bmatrix}\right)^{-1}A_{11}^*[\ I_r, \quad S^*\ ]EM,$

(Z_{10b}) $NF\begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix}\left((A_{11}^*)^{-1}[\ A_{11}^*, \quad A_{21}^*\ ]EMANF\begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix}\right)^{-1}(A_{11}^*)^{-1}[\ A_{11}^*, \quad A_{21}^*\ ]EM,$

(Z_{11}) $NR^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\left(\left(R^{-1|r}T_1\right)^* MANR^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\right)^{-1}\left(R^{-1|r}T_1\right)^* M.$

Dokaz. Iz $A^{\dagger}_{M\circ,\circ N} = (QN)^*(Q(QN)^*)^{-1}((MP)^*P)^{-1}(MP)^*$ sledi da $A^{\dagger}_{M\circ,\circ N}$ predstavlja elemenat klase $A\{1,2\}$ koji ispunjava

$$W_1 = NQ^*, \qquad W_2 = P^*M. \qquad \square$$

Sledeće reprezentacije mogu da se dobiju iz glavnih osobina težinskog Moore-Penroseovog inverza.

Stav 2.6 *Težinski Moore-Penroseov inverz $A^{\dagger}_{\varphi(M,N)}$ za $A \in \mathbb{C}_r^{m\times n}$ može se predsatviti na sledeći način:*

(W_1) $N^{[-1]}\left(G^{-1}{}_{|r}\right)^*\left(\left(R^{-1|r}\right)^* M^{[-1]}AN^{[-1]}\left(G^{-1}{}_{|r}\right)^*\right)^{-1}\left(R^{-1|r}\right)^* M^{[-1]},$

(W_2) $N^{[-1]}\left(G^{-1}{}_{|r}\right)^*\left(\left(R^{-1|r}B\right)^* M^{[-1]}AN^{[-1]}\left(G^{-1}{}_{|r}\right)^*\right)^{-1}\left(R^{-1|r}B\right)^* M^{[-1]},$

(W_3) $N^{[-1]}F\begin{bmatrix} I_r \\ K^* \end{bmatrix}\left(\left(R^{-1|r}\right)^* M^{[-1]}AN^{[-1]}F\begin{bmatrix} I_r \\ K^* \end{bmatrix}\right)^{-1}\left(R^{-1|r}\right)^* M^{[-1]},$

(W_4) $N^{[-1]}\left(G^{-1}{}_{|r}\right)^*\left([\ I_r, \quad K^*\ ]EM^{[-1]}AN^{[-1]}\left(G^{-1}{}_{|r}\right)^*\right)^{-1}[\ I_r, \quad K^*\ ]EM^{[-1]},$

(W_5) $N^{[-1]}\left(G^{-1}{}_{|r}\right)^*\left(U_{|r}M^{[-1]}AN^{[-1]}\left(G^{-1}{}_{|r}\right)^*\right)^{-1}U_{|r}M^{[-1]},$

(W_6) $N^{[-1]}V^{|r}\left(\left(R^{-1|r}\right)^* M^{[-1]}AN^{[-1]}V^{|r}\right)^{-1}\left(R^{-1|r}\right)^* M^{[-1]},$

(W_7) $N^{[-1]}V^{|r}\left(B^*U_{|r}M^{[-1]}AN^{[-1]}V^{|r}\right)^{-1}B^*U_{|r}M^{[-1]},$

(W_8) $N^{[-1]}F\begin{bmatrix} B^* \\ K^* \end{bmatrix}\left(U_{|r}M^{[-1]}AN^{[-1]}F\begin{bmatrix} B^* \\ K^* \end{bmatrix}\right)^{-1}U_{|r}M^{[-1]},$

(W_9) $N^{[-1]}V^{|r}\left([\ B^*, \quad K^*\ ]EM^{[-1]}AN^{[-1]}V^{|r}\right)^{-1}[\ B^*, \quad K^*\ ]EM^{[-1]},$

(W_{10a}) $N^{[-1]}F\begin{bmatrix} I_r \\ T^* \end{bmatrix}\left(A_{11}^*[\ I_r, \quad S^*\ ]EM^{[-1]}AN^{[-1]}F\begin{bmatrix} I_r \\ T^* \end{bmatrix}\right)^{-1}A_{11}^*[\ I_r, \quad S^*\ ]EM^{[-1]},$

(W_{10b}) $N^{[-1]}F\begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix}\left((A_{11}^*)^{-1}[\ A_{11}^*, \quad A_{21}^*\ ]EM^{[-1]}AN^{[-1]}F\begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix}\right)^{-1}$
$\times (A_{11}^*)^{-1}[\ A_{11}^*, \quad A_{21}^*\ ]EM^{[-1]},$

(W_{11}) $N^{[-1]}R^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\left(\left(R^{-1|r}T_1\right)^* M^{[-1]}AN^{[-1]}R^*\begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix}\right)^{-1}\left(R^{-1|r}T_1\right)^* M^{[-1]}.$

Teorema 2.11 *Neka je A kvadratna matrica reda n i ranga r. Tada grupni inverz postoji ako i samo ako su sledeći uslovi (E_i), zavisni od odgovarajućih blokovskih dekompozicija (T_i), ispunjeni:*

(E_1) $G^{-1}{}_{|r}R^{-1|r} = (RG)^{-1}{}_{|r}^{|r}$ *je invertibilno,*

(E_2) $G^{-1}{}_{|r}R^{-1|r}B = (RG)^{-1}{}_{|r}^{|r}B$ *je invertibilno,*

(E_3) $\begin{bmatrix} I_r, & K \end{bmatrix}(RF)^{-1|r} = (RF)^{-1}{}_{|r}^{|r} + K(RF)^{-1}{}_{n-r|}^{|r}$ *je invertibilno,*

(E_4) $(EG)^{-1}{}_{|r}\begin{bmatrix} I_r \\ K \end{bmatrix} = (EG)^{-1}{}_{|r}^{|r} + (EG)^{-1}{}_{|r}^{n-r|}K$ *je invertibilno,*

(E_5) $U^{*|r}G^{-1}{}_{|r} = (UG)^{-1}{}_{|r}^{|r}$ *is invertible,*

(E_6) $V^{*}{}_{|r}R^{-1|r} = (RV)^{-1}{}_{|r}^{|r}$ *je invertibilno,*

(E_7) $V^{*}{}_{|r}U^{*|r}B = (UV)^{*}{}_{|r}^{|r}B$ *je invertibilno,*

(E_8) $\begin{bmatrix} B, & K \end{bmatrix}(UF)^{*}\begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} = \begin{bmatrix} B, & K \end{bmatrix}(UF)^{*|r}$ *je invertibilno,*

(E_9) $\begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix}(EV)^{*}\begin{bmatrix} B \\ K \end{bmatrix} = (EV)^{*}{}_{|r}\begin{bmatrix} B \\ K \end{bmatrix}$ *je invertibilno,*

(E_{10a}) $\begin{bmatrix} I_r, & T \end{bmatrix}(EF)^{*}\begin{bmatrix} I_r \\ S \end{bmatrix}$ *je invertibilno,*

(E_{10b}) $\begin{bmatrix} A_{11}, & A_{12} \end{bmatrix}(EF)^{*}\begin{bmatrix} A_{11} \\ A_{12} \end{bmatrix}$ *je invertibilno,*

(E_{11}) $\begin{bmatrix} I_r, & T_1^{-1}T_2 \end{bmatrix}\begin{bmatrix} T_1 \\ \mathbb{O} \end{bmatrix} = T_1$ *je invertibilno* [8].

Takođe, grupni inverz, ako postoji, poseduje sledeće blokovske reprezentacije:

$(A^1_{\#})$ $A^{\#} = R^{-1|r}\left((RG)^{-1}{}_{|r}^{|r}\right)^{-2}G^{-1}{}_{|r},$

$(A^2_{\#})$ $A^{\#} = R^{-1|r}B^{-1}\left((RG)^{-1}{}_{|r}^{|r}\right)^{-2}G^{-1}{}_{|r},$

$(A^3_{\#})$ $A^{\#} = R^{-1|r}\left((RF)^{-1}{}_{|r}^{|r} + K(RF)^{-1}{}_{n-r|}^{|r}\right)^{-2}\left(F^{*}{}_{|r} + KF^{*}_{n-r|}\right),$

$(A^4_{\#})$ $A^{\#} = \left(E^{*|r} + E^{*n-r|}K\right)\left((EG)^{-1}{}_{|r}^{|r} + (EG)^{-1}{}_{|r}^{n-r|}K\right)^{-2}G^{-1}{}_{|r},$

$(A^5_{\#})$ $A^{\#} = U^{*|r}\left((UG)^{-1}{}_{|r}^{|r}\right)^{-2}G^{-1}{}_{|r},$

$(A^6_{\#})$ $A^{\#} = R^{-1|r}\left((RV)^{-1}{}_{|r}^{|r}\right)^{-2}V^{*}{}_{|r},$

$(A^7_{\#})$ $A^{\#} = U^{*|r}B^{-1}\left((UV)^{*}{}_{|r}^{|r}\right)^{-2}V^{*}{}_{|r},$

$(A^8_{\#})$ $A^{\#} = U^{*|r}\left(\begin{bmatrix} B, & K \end{bmatrix}(UF)^{*}\right)^{-2}\begin{bmatrix} B, & K \end{bmatrix}F^{*},$

$(A^9_{\#})$ $A^{\#} = E^{*}\begin{bmatrix} B \\ K \end{bmatrix}\left((EV)^{*}{}_{|r}\begin{bmatrix} B \\ K \end{bmatrix}\right)^{-2}V^{*}{}_{|r},$

$(A_{\#}^{10a})\quad A^{\#} = E^* \begin{bmatrix} I_r \\ S \end{bmatrix} A_{11} \left(\begin{bmatrix} I_r, & T \end{bmatrix} (EF)^* \begin{bmatrix} I_r \\ S \end{bmatrix} A_{11} \right)^{-2} \begin{bmatrix} I_r, & T \end{bmatrix} F^*,$

$(A_{\#}^{10b})\quad A^{\#} = E^* \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} A_{11} \left(\begin{bmatrix} A_{11}, & A_{12} \end{bmatrix} (EF)^* \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} \right)^{-2} \begin{bmatrix} A_{11}, & A_{12} \end{bmatrix} F^*,$

$(A_{\#}^{11})\quad A^{\#} = R^{-1} \begin{bmatrix} T_1^{-1} & T_1^{-2}T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix} R = R^{-1}{}_{|r} \begin{bmatrix} T_1^{-1}, & T_1^{-1}T_2 \end{bmatrix} R.$

Dokaz. Sledi neposredno iz sledećeg: Ako je $A = PQ$ potpuna-rang faktorizacija za A, tada $A^{\#}$ postoji ako i samo ako je QP invertibilna matrica, i $A^{\#} = P(QP)^{-2}Q$. □

Metode za izračunavanje grupnog inverza mogu se implementirati na sledeći način:

```
Group1[b_]:=
Block[{a=b,ran,r,g,rg,pom1,pom2},
   {r,g}=t1[a]; ran=rank[a];
   rg=r.g;
   rg=Inverse[rg];
   rg=Take[rg,ran];
   rg=Transpose[Take[Transpose[rg],ran]];
   If[rank[rg]==ran,
        pom1=Transpose[Take[Transpose[Inverse[r]],ran]];
        pom2=Take[Inverse[g],ran];
        Return[pom1.MatrixPower[Inverse[rg],2].pom2],
        Return["Group inverse does not exists"];
   ];
]

Group3[b_]:=
Block[{a=b,ran,r,g,pom1,pom2},
   {r,f}=t3[a]; ran=rank[a];
   pom1=Transpose[Take[Transpose[Inverse[r]],ran]];
   pom2=Take[r.a.f,ran].Hermit[f];
   pom3=pom2.pom1;
   If[rank[pom3]==ran,
       Return[pom1.MatrixPower[Inverse[pom3],2].pom2],
       Return["Group inverse does not exists"];
   ];
]

Group4[b_]:=
Block[{a=b,ran,r,g,k,pom1,pom2},
   {e,g}=t3[a]; ran=rank[a];
```

```
    pom1=Take[Inverse[g],ran];
    pom2=Hermit[e].Transpose[Take[Transpose[e.a.g],ran]];
    pom3=pom2.pom1;
    If[rank[pom3]==ran,
        Return[pom2.MatrixPower[Inverse[pom3],2].pom1],
        Return["Group inverse does not exists"];
    ];
]
```

Primer 2.9 Za PQ faktorizaciju matrice $A = \begin{bmatrix} 9 & 6 & 9 \\ 6 & 6 & 6 \\ 9 & 6 & 9 \end{bmatrix}$ imamo redom

$$R = \begin{bmatrix} \frac{1}{9} & 0 & 0 \\ -\frac{1}{3} & \frac{1}{2} & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & -\frac{2}{3} & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} 9 & 0 & 0 \\ 6 & 2 & 0 \\ 9 & 0 & 1 \end{bmatrix}, \quad G^{-1} = \begin{bmatrix} 1 & \frac{2}{3} & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P = R^{-1} \begin{bmatrix} I_2 \\ \mathbb{O} \end{bmatrix} = \begin{bmatrix} 9 & 0 & 0 \\ 6 & 2 & 0 \\ 9 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 6 & 2 \\ 9 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} I_2 & \mathbb{O} \end{bmatrix} G^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & \frac{2}{3} & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{2}{3} & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Primer 2.10 Data je matrica $A = \begin{bmatrix} -1 & 1 & 3 & 5 & 7 \\ 1 & 0 & -2 & 0 & 4 \\ 1 & 1 & -1 & 5 & 15 \\ -1 & 2 & 4 & 10 & 18 \end{bmatrix}$.

Iz $\{r1, f1\} = t3[a]$ dobijamo

$$R_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & -2 & 1 & 0 \\ -2 & -1 & 0 & 1 \end{bmatrix}, \quad F_1 = I_5.$$

Primenom $\{r2, f2\} = t3[Transpose[r1.a.f1]]$ dobijamo

$$R_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 2 & -1 & 1 & 0 & 0 \\ 0 & -5 & 0 & 1 & 0 \\ -4 & -11 & 0 & 0 & 1 \end{bmatrix}, \quad F_2 = I_4.$$

Sada je $R = F_2^T R_1$, $G = F_1 R_2^T$, i

$$R^{-1|2} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 1 & 1 \\ -1 & 2 \end{bmatrix}, \quad G^{-1}{}_{|2} = \begin{bmatrix} 1 & 0 & -2 & 0 & 4 \\ 0 & 1 & 1 & 5 & 11 \end{bmatrix}.$$

Primenom formule $(A_\dagger^1)$ dobijamo

$$A^\dagger = \begin{bmatrix} -\frac{169}{6720} & \frac{67}{2240} & \frac{233}{6720} & -\frac{137}{6720} \\ \frac{1}{128} & -\frac{1}{128} & -\frac{1}{128} & \frac{1}{128} \\ \frac{781}{13440} & -\frac{330}{4480} & -\frac{1037}{13440} & \frac{653}{13440} \\ \frac{5}{128} & -\frac{5}{128} & -\frac{5}{128} & \frac{5}{128} \\ -\frac{197}{13440} & \frac{151}{4480} & \frac{709}{13440} & \frac{59}{13440} \end{bmatrix}.$$

Primer 2.11 Za $A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}$ dobijamo

$$A_{11}^{-1} = \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}, \quad T = \begin{bmatrix} -1 & -2 \\ -1 & -3 \end{bmatrix}.$$

Primenom opisanih teorema i posledica za

$$W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 4 & 2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & -1 \\ 0 & -1 & 0 & 0 & -2 & 1 \end{bmatrix},$$

dobijamo:

$$A^{(1,2)} = \begin{bmatrix} -\frac{10}{17} & \frac{44}{51} & -\frac{10}{17} & -\frac{10}{51} & \frac{88}{51} & -\frac{44}{51} \\ \frac{13}{17} & -\frac{47}{51} & \frac{13}{17} & \frac{13}{51} & -\frac{94}{51} & \frac{47}{51} \\ -\frac{5}{17} & \frac{22}{51} & -\frac{5}{17} & -\frac{5}{51} & \frac{44}{51} & -\frac{22}{51} \\ \frac{6}{17} & -\frac{2}{17} & \frac{6}{17} & \frac{2}{17} & -\frac{4}{17} & \frac{2}{17} \end{bmatrix},$$

$$A^{(1,2,3)} = \begin{bmatrix} -\frac{2}{17} & -\frac{3}{17} & \frac{1}{17} & -\frac{1}{17} & \frac{3}{17} & \frac{2}{17} \\ \frac{19}{102} & \frac{10}{51} & -\frac{1}{102} & \frac{1}{102} & -\frac{10}{51} & -\frac{19}{102} \\ -\frac{1}{17} & -\frac{3}{34} & \frac{1}{34} & -\frac{1}{34} & \frac{3}{34} & \frac{1}{17} \\ \frac{7}{51} & \frac{2}{51} & \frac{5}{51} & -\frac{5}{51} & -\frac{2}{51} & -\frac{7}{51} \end{bmatrix},$$

$$A^{(1,2,4)} = \begin{bmatrix} -\frac{11}{17} & \frac{43}{51} & -\frac{11}{17} & -\frac{11}{51} & \frac{86}{51} & -\frac{43}{51} \\ \frac{7}{17} & -\frac{32}{51} & \frac{7}{17} & \frac{7}{51} & -\frac{64}{51} & \frac{32}{51} \\ \frac{4}{17} & -\frac{11}{51} & \frac{4}{17} & \frac{4}{51} & -\frac{22}{51} & \frac{11}{51} \\ \frac{1}{17} & \frac{10}{51} & \frac{1}{17} & \frac{1}{51} & \frac{20}{51} & -\frac{10}{51} \end{bmatrix},$$

$$A^{\dagger} = \begin{bmatrix} -\frac{5}{34} & -\frac{3}{17} & \frac{1}{34} & -\frac{1}{34} & \frac{3}{17} & \frac{5}{34} \\ \frac{4}{51} & \frac{13}{102} & -\frac{5}{102} & \frac{5}{102} & -\frac{13}{102} & -\frac{4}{51} \\ \frac{7}{102} & \frac{5}{102} & \frac{1}{51} & -\frac{1}{51} & -\frac{5}{102} & -\frac{7}{102} \\ \frac{1}{17} & -\frac{1}{34} & \frac{3}{34} & -\frac{3}{34} & \frac{1}{34} & -\frac{1}{17} \end{bmatrix}.$$

Primer 2.12 Uočimo iste matrice. Koristeći proširenu Gauss-Jordanovu transformaciju, dobijamo sledeću redukovanu row-echelon formu za A:

$$B = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & -1 & -3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Matrica B je dobijena koristeći permutacionu matricu $E = I_4$, i regularnu matricu

$$R = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Primenom istih matrica W_1 i W_2 dobijamo:

$$A^{(1,2)} = \begin{bmatrix} -\frac{2}{51} & -\frac{6}{17} & \frac{10}{17} & \frac{10}{51} & -\frac{14}{51} & \frac{8}{17} \\ \frac{23}{51} & \frac{1}{17} & -\frac{13}{17} & -\frac{13}{51} & \frac{8}{51} & -\frac{7}{17} \\ -\frac{1}{51} & -\frac{3}{17} & \frac{5}{17} & \frac{5}{51} & -\frac{7}{51} & \frac{4}{17} \\ \frac{14}{17} & -\frac{10}{17} & -\frac{6}{17} & -\frac{2}{17} & -\frac{4}{17} & \frac{2}{17} \end{bmatrix},$$

$$A^{(1,2,4)} = \begin{bmatrix} -\frac{13}{51} & -\frac{3}{17} & \frac{11}{17} & \frac{11}{51} & -\frac{10}{51} & \frac{7}{17} \\ -\frac{1}{51} & \frac{5}{17} & -\frac{7}{17} & -\frac{7}{51} & \frac{11}{51} & -\frac{6}{17} \\ \frac{14}{51} & -\frac{2}{17} & -\frac{4}{17} & -\frac{4}{51} & -\frac{1}{51} & -\frac{1}{17} \\ \frac{29}{51} & -\frac{9}{17} & -\frac{1}{17} & -\frac{1}{51} & -\frac{13}{51} & \frac{4}{17} \end{bmatrix}.$$

Primer 2.13 Za matricu$A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 1 & 1 & -2 & -5 \end{bmatrix}$ dobijamo

$$R = \begin{bmatrix} -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 2 & -1 & 0 & 1 \end{bmatrix}, \quad T_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} -1 & -2 \\ -1 & -3 \end{bmatrix}, \quad E = I_4.$$

Koristeći matrice

$$W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 4 & 2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 3 & 1 & 3 & 1 \\ 0 & -1 & 0 & 0 \end{bmatrix},$$

dobija se:

$$A^{(1,2)} = \begin{bmatrix} -\frac{114}{221} & -\frac{38}{221} & \frac{44}{221} & -\frac{22}{221} \\ \frac{138}{221} & \frac{46}{221} & -\frac{30}{221} & \frac{15}{221} \\ -\frac{57}{221} & -\frac{19}{221} & \frac{22}{221} & -\frac{11}{221} \\ \frac{48}{221} & \frac{16}{221} & \frac{28}{221} & -\frac{14}{221} \end{bmatrix},$$

$$A^{(1,2,3)} = \begin{bmatrix} -\frac{10}{51} & -\frac{6}{17} & \frac{8}{51} & \frac{2}{51} \\ \frac{13}{51} & \frac{20}{51} & -\frac{7}{51} & -\frac{2}{17} \\ -\frac{5}{51} & -\frac{3}{17} & \frac{4}{51} & \frac{1}{51} \\ \frac{2}{17} & \frac{4}{51} & \frac{2}{51} & -\frac{8}{51} \end{bmatrix},$$

$$A^{(1,2,4)} = \begin{bmatrix} -\frac{120}{221} & -\frac{40}{221} & \frac{2}{13} & -\frac{1}{13} \\ \frac{81}{221} & \frac{27}{221} & -\frac{2}{13} & \frac{1}{13} \\ \frac{3}{17} & \frac{1}{17} & 0 & 0 \\ -\frac{3}{221} & -\frac{1}{221} & \frac{2}{13} & -\frac{1}{13} \end{bmatrix},$$

$$A^{\dagger} = \begin{bmatrix} -\frac{11}{51} & -\frac{6}{17} & \frac{7}{51} & \frac{4}{51} \\ \frac{7}{51} & \frac{13}{51} & -\frac{2}{17} & -\frac{1}{51} \\ \frac{4}{51} & \frac{5}{51} & -\frac{1}{51} & -\frac{1}{17} \\ \frac{1}{51} & -\frac{1}{17} & \frac{4}{51} & -\frac{5}{51} \end{bmatrix}.$$

$$A^{\#} = \begin{bmatrix} -5 & 4 & 1 & -2 \\ -21 & 17 & 4 & -9 \\ 16 & -13 & -3 & 7 \\ -11 & 9 & 2 & -5 \end{bmatrix}.$$

2.1.4 Metoda pregradjivanja

Prvi metod pregrađivanja (partitioning method) potiče og Grevile [33]. Ovaj metod predstavlja rekurzivni algoritam za izračunavanje Moore-Penroseovog inverza. U ovom metodu se koristi izraz koji povezuje Moore-Penroseov inverz matrice $[A|a]$ (matrice A sa dodatom kolonom a) i Moore-Penroseov inverz matrice A.

Algoritam 2.1 *Posmatrajmo matricu A dimenzije $m \times n$. Neka je A_i podmatrica matrice A sastavljena iz prvih i kolona matrice A. Ako i-tu kolonu matrice A označimo sa a_i, onda se A_i može predstaviti kao $A_i = \left[\ A_{i-1}|a_i\ \right]$.*

Korak 1. Početne vrednosti:

$$A_1^{\dagger} = a_1^{\dagger} = \begin{cases} \frac{1}{a_1^* a_1} a_1^*, & a_1 \neq 0, \\ a_1^*, & a_1 = 0 \end{cases}$$

Korak 2. Rekurzivni korak: Za svaku kolonu a_i, $2 \leq i \leq n$ matrice A izračunati

$$A_i^{\dagger} = \begin{bmatrix} A_{i-1}^{\dagger} - d_i b_i^* \\ b_i^* \end{bmatrix},$$

gde je

Korak 2.1. $d_i = A_{i-1}^{\dagger} a_i$,

Korak 2.2. $c_i = a_i - A_{i-1} d_i = (I - A_{i-1} A_{i-1}^{\dagger}) a_i$,

Korak 2.3. $b_i = \begin{cases} \frac{1}{c_i^* c_i} c_i, & c_i \neq 0 \\ \frac{1}{1 + d_i^* d_i} (A_{i-1}^{\dagger})^* d_i, & c_i = 0 \end{cases}$

Korak 3. Kritetijum zaustavljanja: $A^\dagger = A_n^\dagger$.

Grevile-ov algoritam ocenjen je kao relativno komplikovan i numerički stabilan u [58]. Glavni problem nastao u rekurzivnoj implementaciji algoritma potiče iz višestrukog ponovnog izračunavanja pojedinih minora. U *Koraku 2*, za svako $i \in \{2, \dots, n\}$, Moore-Penrose-ov inverz $A_i^\dagger$ se mora izračunati $n - i + 1$ puta. Tim pre i u *Koraku 2.1* i *Koraku 2.3*, takođe je neophodno generisati inverz $A_{i-1}^\dagger$ prilikom izračunavanja vrednosti d_i i b_i. Ukupan broj različitih vrednosti koje su proizvod izračunavanja je relativno mali, ali ove vrednosti se moraju ponovo izračunavati mnogo puta. Da bi izbegli ovaj problem, možemo upotrebiti mogućnosti programskog paketa MATHEMATICA u definisanju funkcija koje pamte vrednosti koje su generisane u među koracima (o MATHEMATICA videti, npr [83, 84]).

Prvo su opisane neke pomoćne procedure.

A. Procedura koja generiše j-tu kolonu a_j matrice A:

```
Col[a_List,j_]:=Transpose[{Transpose[a][[j]]}]
```

B. Podmatrica $A_j = [\ a_1, \cdots a_j\ ]$ koja sadrži prvih $j \le n$ kolona matrice $A = A_n = [\ a_1, \cdots a_n\ ]$ može se pronaći na sledeći način:

```
TakeFirstI[a_List,j_]:= Return[Transpose[Take[Transpose[a],j]]];
```

Implementacija *Koraka 2.1*.

```
DD[a_List,i_]:=DD[a,i]=
   Module[{s={}},
     s=Simplify[A[a,i-1].Col[a,i]];
     If[Length[s]==1,  Return[s[[1]]],  Return[s] ];
   ]
```

Implementacija *Koraka 2.2.*

```
CC[a_List,i_]:=CC[a,i]=
   Module[{s={},vr},
      vr=Variables[DD[a,i]];
      If[vr=!={},
         If[Length[Dimensions[DD[a,i]]]==1,
              s=Col[a,i]-TakeFirstI[a,i-1]DD[a,i],
              s=Col[a,i]-TakeFirstI[a,i-1].DD[a,i]
         ],
         If[Length[DD[a,i]]==0,
              s=Col[a,i]-TakeFirstI[a,i-1]DD[a,i],
              s=Col[a,i]-TakeFirstI[a,i-1].DD[a,i]
```

```
      ]
    ];
    Return[Simplify[s]]
  ]
```

Implementacija *Koraka 2.3.*

```
B[a_List,i_]:=B[a,i]=
  Module[{nul,m1,j,k,n1,s={}},
  {m1,n1}=Dimensions[CC[a,i]];
   nul=Table[0,{j,1,m1},{k,1,n1}];
   If[CC[a,i]=!=nul,
    s=(1/(Transpose[CC[a,i]].CC[a,i])[[1,1]])CC[a,i],
    If[Length[Dimensions[DD[a,i]]]==1,
      s=(1/(1+DD[a,i]DD[a,i]))Transpose[{A[a,i-1]}]DD[a,i],
      s=(1/(1+Transpose[DD[a,i]].DD[a,i])[[1,1]])
           Transpose[A[a,i-1]].DD[a,i]
   ]];
   Return[Simplify[s]];
  ]
```

Implementacija *Koraka 1, Koraka 2 i Koraka 3.*

```
A[a_List,i_]:=A[a,i]=
  Module[{b=a,vr},
   If[i==1,
      If[Col[a,i]==Col[a,i]*0, b=Transpose[a][[1]],
          b=(1/(Transpose[a][[i]].Col[a,i])[[1]])Transpose[a][[1]]
      ],
      vr=Variables[DD[a,i]];
      If[vr=!={},
         If[Length[Dimensions[DD[a,i]]]==1,
            b={A[a,i-1]}-DD[a,i]Transpose[B[a,i]],
            b=A[a,i-1]-DD[a,i].Transpose[B[a,i]]
         ],
         If[Length[DD[a,i]]==0,
           b={A[a,i-1]}-DD[a,i]Transpose[B[a,i]],
           b=A[a,i-1]-DD[a,i].Transpose[B[a,i]]
      ]  ];
      b=Append[b,Transpose[B[a,i]][[1]]]
   ];
   Return[Simplify[b]]
  ]

Partitioning[a_List]:=
```

```
Block[{m,n}, {m,n}=Dimensions[a]; A[a,n] ];
```

Kao što je opisano u [83, 84], obično je brže naći određene vrednosti nego ih ponovo izračunavati, ali je potrebno malo više memorijskog prostora da bi se te prethodno izračunate vrednosti memorisale. Na osnovu numeričkih rezultata uočeno je da je rekurzivna implementacija bazirana na memorisanju prethodno izračunatih vrednosti je brža nego iterativna implementacija. Čak, sta više, rekurzivna implementacija algoritma je jednostavnija i prirodnija, i vrlo je numerički atraktivna.

Primer 2.14 Sledeći primer poznat je kao S_{11} iz [85]:

$$S11 = \begin{pmatrix} a+1 & a & a & a & a & a & a & a & a & a & a+1 \\ a & a-1 & a & a & a & a & a & a & a & a & a \\ a & a & a+1 & a & a & a & a & a & a & a & a \\ a & a & a & a-1 & a & a & a & a & a & a & a \\ a & a & a & a & a+1 & a & a & a & a & a & a \\ a & a & a & a & a & a-1 & a & a & a & a & a \\ a & a & a & a & a & a & a+1 & a & a & a & a \\ a & a & a & a & a & a & a & a-1 & a & a & a \\ a & a & a & a & a & a & a & a & a+1 & a & a \\ a & a & a & a & a & a & a & a & a & a-1 & a \\ a+1 & a & a & a & a & a & a & a & a & a & a+1 \end{pmatrix}.$$

Funkcija *Partitioning*[$S11$] daje dobro poznat inverz $S_{11}^{\dagger}$:

$$\text{Out[1]=} \begin{pmatrix} \frac{1-a}{4} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & \frac{1-a}{4} \\ \frac{a}{2} & -1-a & a & -a & a & -a & a & -a & a & -a & a \\ -\frac{a}{2} & a & 1-a & a & -a & a & -a & a & -a & a & -a \\ \frac{a}{2} & -a & a & -1-a & a & -a & a & -a & a & -a & a \\ -\frac{a}{2} & a & -a & a & 1-a & a & -a & a & -a & a & -a \\ \frac{a}{2} & -a & a & -a & a & -1-a & a & -a & a & -a & a \\ -\frac{a}{2} & a & -a & a & -a & a & 1-a & a & -a & a & -a \\ \frac{a}{2} & -a & a & -a & a & -a & a & -1-a & a & -a & a \\ -\frac{a}{2} & a & -a & a & -a & a & -a & a & 1-a & a & -a \\ \frac{a}{2} & -a & a & -a & a & -a & a & -a & a & -1-a & a \\ \frac{1-a}{4} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & \frac{1-a}{4} \end{pmatrix}$$

Više primera sa funkcijom *Partitioning*[] mogu se naći u daljem tekstu prilikom upoređenja drugih metoda sa opisanim.

2.1.5 Determinantska reprezentacija

Pod determinantskom reprezentacijom generalisanih inverza matrice A podrazumevamo izračunavanje elemenata tih inverza pomoću minora matrice A. Problem determinantske reprezentacije generalisanih inverza razmatran je još u prvim radovima Moorea [15]. Arghiriade i Dragomir su 1963 izveli determinantsku reprezentaciju Moore-Penrose inverza za matrice potpunog ranga. Uopštenje tog rezultata na matrice nepotpunog ranga dao je R. Gabriel 1965. On je kasnije uopštio svoj rezultat posmatrajući generalisane inverze u različitim algebarskim strukturama. Bapat, Rao i Manjunatha su u novije vreme proučavali determinantsku reprezentaciju generalisanih inverza u integralnim domenima [3, 50, 51].

Sledi kratak opis glavnih rezultata iz pomenutih radova, koji se odnose na kompleksne matrice.

Moore je uveo sledeće jednakosti, kojima se element $a_{ij}^{\dagger}$ matrice $A^{\dagger}$ predstavlja u obliku razlomka sume determinanti.

Teorema 2.12 *Za $A \in \mathbb{C}_r^{m\times n}$, (i,j)-ti element inverza $A^{\dagger}$ je dat sa*

$$a_{ij}^{\dagger} = \frac{\sum\limits_{\substack{i_2<\ldots<i_r\\ j_2<\ldots<j_r}} A^*\begin{pmatrix} i & i_2 & \ldots & i_r\\ j & j_2 & \ldots & j_r\end{pmatrix} A\begin{pmatrix} j_2 & \ldots & j_r\\ i_2 & \ldots & i_r\end{pmatrix}}{\sum\limits_{\substack{s_1<\ldots<s_r\\ t_1<\ldots<t_r}} A\begin{pmatrix} s_1 & \ldots & s_r\\ t_1 & \ldots & t_r\end{pmatrix} A^*\begin{pmatrix} t_1 & \ldots & t_r\\ s_1 & \ldots & s_r\end{pmatrix}}.$$

Problem da se definiše Moore-Penroseov inverz pomoću opšteg algebarskog komplementa prvi put je razmatrao Arghiriade. Arghiriade i Dragomir su generalisali pojam algebarskog komplementa i izveli determinantsku reprezentaciju Moore-Penroseovog pseudoinversa za matrice potpunog ranga. U tom radu nisu citirali Mooreov rezultat.

Teorema 2.13 *Za zadatu matricu potpunog ranga $A \in \mathbb{C}^{m\times n}$, generalisani algebarski komplement odgovarajući elementu a_{ij}, jednak je sa*

$$A_{ij}^{\dagger} = \begin{cases} \sum\limits_{\beta_1<\ldots<\beta_m} \overline{A}\begin{pmatrix} 1 & \ldots & \ldots & \ldots & m\\ \beta_1 & \ldots & j & \ldots & \beta_m\end{pmatrix} A_{ij}\begin{pmatrix} 1 & \ldots & \ldots & \ldots & m\\ \beta_1 & \ldots & j & \ldots & \beta_m\end{pmatrix}, & m\le n,\\ \sum\limits_{\alpha_1<\ldots<i<\ldots<\alpha_n} \overline{A}\begin{pmatrix} \alpha_1 & \ldots & \alpha_n & & \\ 1 & \ldots & \ldots & \ldots & n\end{pmatrix} A_{ij}\begin{pmatrix} \alpha_1 & \ldots & i & \ldots & \alpha_n\\ 1 & \ldots & \ldots & \ldots & n\end{pmatrix}, & n\le m.\end{cases}$$

Norma za A je

$$\|A\| = \begin{cases} \sum\limits_{1\le\beta_1<\ldots<\beta_m\le n} \overline{A}\begin{pmatrix} 1 & \ldots & m\\ \beta_1 & \ldots & \beta_m\end{pmatrix} A\begin{pmatrix} 1 & \ldots & m\\ \beta_1 & \ldots & \beta_m\end{pmatrix}, & m\le n,\\ \sum\limits_{1\le\alpha_1<\ldots<\alpha_n\le m} \overline{A}\begin{pmatrix} \alpha_1 & \ldots & \alpha_n\\ 1 & \ldots & n\end{pmatrix} A\begin{pmatrix} \alpha_1 & \ldots & \alpha_n\\ 1 & \ldots & n\end{pmatrix}, & n\le m,\end{cases}$$

dok je (i,j)-ti element Moore-Penroseovog inverza

$$A^{\dagger} = \begin{cases} A^*(AA^*)^{-1}, & m\le n,\\ (A^*A)^{-1}A^*, & n\le m\end{cases}$$

jednak

$$a_{ij}^{\dagger} = \frac{1}{\|A\|}A_{ij}^{\dagger}, \quad \begin{pmatrix} 1\le i\le n\\ 1\le j\le m\end{pmatrix}.$$

R. Gabriel je dobio isti rezultat, uvodeći eksplicitnu formulu za (i,j)-ti element matričnog izraza $\frac{A^*\cdot \mathrm{adj}AA^*}{\det(AA^*)}$, odnosno $\frac{\mathrm{adjA^*A\cdot A^*}}{\det(A^*A)}$.

Teorema 2.14 *Element na i-toj vrsti j-toj koloni Moore-Penroseovog pseudoinverza date matrice* $A \in \mathbb{C}_r^{m\times n}$ *može da se reprezentuje pomoću kvadratnih minora na sledeći način:*

$$a_{ij}^{\dagger} = \frac{A_{ij}^{\dagger}}{\|A\|} = \frac{\sum\limits_{\substack{1\le\beta_1<\ldots<\beta_r\le n\\ 1\le\alpha_1<\ldots<\alpha_r\le m}} \overline{A}\begin{pmatrix}\alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix} A_{ji}\begin{pmatrix}\alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix}}{\sum\limits_{\substack{1\le\delta_1<\ldots<\delta_r\le n\\ 1\le\gamma_1<\ldots<\gamma_r\le m}} A\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}\overline{A}\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}}.$$

Matricu sa elementima jednakim $A_{ij}^{\dagger}$ označavamo sa $\mathrm{adj}^{\dagger}(\mathrm{A})$, i nazivamo *generalisana ađungovana matrica* za A.

Teorema 2.15 *Neka su* $M \in \mathbb{C}^{m\times m}$, $N \in \mathbb{C}^{n\times n}$ *pozitivno definisane matrice, i neka je* $A = PQ$ *potpuna rang faktorizacija za* A, *tako da je* $rank(P^*MP) = rank(QNQ^*) = rank(MAN) = r$. *Element težinskog Moore-Penroseovog inverza* $A_{M,N}^{\dagger}$, *koji se nalazi u preseku i-te vrste i j-te kolone, može da se predstavi pomoću kvadratnih minora matrica* A *i* MAN *kako sledi:*

$$(a_{M,N}^{\dagger})_{ij} = \frac{\sum\limits_{\substack{1\le\beta_1<\ldots<\beta_r\le n\\ 1\le\alpha_1<\ldots<\alpha_r\le m}} \overline{(MAN)}\begin{pmatrix}\alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix} A_{ji}\begin{pmatrix}\alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix}}{\sum\limits_{\substack{1\le\delta_1<\ldots<\delta_r\le n\\ 1\le\gamma_1<\ldots<\gamma_r\le m}} A\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}\overline{(MAN)}\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}}.$$

Teorema 2.16 *Ako je* $A = PQ$ *potpuna rang faktorizacija matrice* $A \in \mathbb{C}_r^{m\times n}$, *i* $W_1 \in \mathbb{C}^{n\times r}$, $W_2 \in \mathbb{C}^{r\times m}$ *takve da je* $rank(QW_1) = rank(W_2P) = rank(W_1W_2) = rank(A)$, *tada se element na i-toj vrsti i j-toj koloni* $\{i,j,k\}$*-inverza izračunava sa:*

$$a_{ij}^{(1,2)} = \frac{\sum\limits_{\substack{1\le\beta_1<\ldots<\beta_r\le n\\ 1\le\alpha_1<\ldots<\alpha_r\le m}} (W_1W_2)^T\begin{pmatrix}\alpha_1 & \ldots & i & \ldots & \alpha_r\\ \beta_1 & \ldots & j & \ldots & \beta_r\end{pmatrix} A_{ji}\begin{pmatrix}\alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix}}{\sum\limits_{\substack{1\le\delta_1<\ldots<\delta_r\le n\\ 1\le\gamma_1<\ldots<\gamma_r\le m}} A\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}(W_1W_2)^T\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}}.$$

$$a_{ij}^{(1,2,3)} = \frac{\sum\limits_{\substack{1\le\beta_1<\ldots<\beta_r\le n\\ 1\le\alpha_1<\ldots<\alpha_r\le m}} (W_1P^*)^T\begin{pmatrix}\alpha_1 & \ldots & i & \ldots & \alpha_r\\ \beta_1 & \ldots & j & \ldots & \beta_r\end{pmatrix} A_{ji}\begin{pmatrix}\alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix}}{\sum\limits_{\substack{1\le\delta_1<\ldots<\delta_r\le n\\ 1\le\gamma_1<\ldots<\gamma_r\le m}} A\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}(W_1P^*)^T\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}};$$

$$a_{ij}^{(1,2,4)} = \frac{\sum\limits_{\substack{1\le \beta_1<\ldots<\beta_r\le n\\ 1\le \alpha_1<\ldots<\alpha_r\le m}} (Q^*W_2)^T\begin{pmatrix} \alpha_1 & \ldots & i & \ldots & \alpha_r\\ \beta_1 & \ldots & j & \ldots & \beta_r\end{pmatrix} A_{ji}\begin{pmatrix} \alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix}}{\sum\limits_{\substack{1\le \delta_1<\ldots<\delta_r\le n\\ 1\le \gamma_1<\ldots<\gamma_r\le m}} A\begin{pmatrix} \gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}(Q^*W_2)^T\begin{pmatrix} \gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}}.$$

Teorema 2.17 *Grupni inverz* $A^{\#} = \left(a_{ij}^{\#}\right)$ *matrice* $A \in \mathbb{C}_r^{n\times n}$ *ima ovakvu determinantsku reprezentaciju:*

$$a_{ij}^{\#} = \frac{\sum\limits_{\substack{1\le \alpha_1<\ldots<\alpha_r\le n\\ 1\le \beta_1<\ldots<\beta_r\le n}} A^T\begin{pmatrix} \alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix} A_{ji}\begin{pmatrix} \alpha_1 & \ldots & j & \ldots & \alpha_r\\ \beta_1 & \ldots & i & \ldots & \beta_r\end{pmatrix}}{\sum\limits_{\substack{1\le \gamma_1<\ldots<\gamma_r\le n\\ 1\le \delta_1<\ldots<\delta_r\le n}} A^T\begin{pmatrix} \gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix} A\begin{pmatrix} \gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}}.$$

Pravougaone determinante i generalisani inverzi

Definicija 2.1 *Determinata matrice* $A \in \mathbb{C}_r^{m\times n}$ *je funkcija* $\det_{(\epsilon,\mathrm{p})} : \mathbb{C}^{\mathrm{m}\times\mathrm{n}} \mapsto \mathbb{C}$*, definisana sa:*

$$\det_{(\epsilon,p)}(A) = \begin{cases} \sum\limits_{\substack{1\le \alpha_1<\ldots<\alpha_p\le m\\ 1\le \beta_1<\ldots<\beta_p\le n}} \epsilon^{(\alpha_1+\ldots+\alpha_p)+(\beta_1+\ldots+\beta_p)} A\begin{pmatrix} \alpha_1 & \ldots & \alpha_p\\ \beta_1 & \ldots & \beta_p\end{pmatrix}, & p \le \min\{m,n\},\\ 0, & p > \min\{m,n\}.\end{cases}$$

Za $\epsilon = 1$, i $p = r$, dobijamo Stojakovićevu deteminatu, označenu sa $\det_{(S,p)}(A)$. Slično, za $\epsilon = -1$ i $p = r$, dobijamo determinantu uvedenu od M. Radića (označenu sa $\det_{(R,p)}(A)$). U ovom radu se ove determinante kraće nazivaju *pravougaone determinante.*

Definicija 2.2 *Za* $A \in \mathbb{C}_r^{m\times n}$ *generalisani inverz* $A_{(\epsilon,p)}^{-1}$ *matrice A je matrica sa* (i,j)*-tim elementom*

$$\left(A_{(\epsilon,p)}^{-1}\right)_{ij} = \frac{A_{ij}^{(\epsilon,p)}}{\det_{(\epsilon,\mathrm{p})}(\mathrm{A})}.$$

Ovde je $1 \le p \le rank(A) \le \min\{m,n\}$ *najveći ceo broj, takav da je* $\det_{\mathrm{p}}^{\epsilon}(\mathrm{A}) \ne 0$ *(označen sa* $r_\epsilon(A)$*), i* $A_{ij}^{(\epsilon,p)}$ *je generalisani algebarski komplement reda p odgovarajući elementu* a_{ji}*, koji se definiše se na sledeći način:*

$$A_{ij}^{(\epsilon,p)} = \sum_{\substack{1\le j_1<\ldots<j<\ldots<j_p\le n\\ 1\le i_1<\ldots<i<\ldots<i_p\le m}} \epsilon^{(i_1+\ldots+i_p)+(j_1+\ldots+j_p)} A_{ji}\begin{pmatrix} j_1 & \ldots & j & \ldots & j_p\\ i_1 & \ldots & i & \ldots & i_p\end{pmatrix}.$$

Matricu $\mathrm{adj}^{(\epsilon,\mathrm{p})}(\mathrm{A}) = \left(\mathrm{A}_{\mathrm{ij}}^{(\epsilon,\mathrm{p})}\right), \begin{pmatrix} 1 \le i \le n \\ 1 \le j \le m \end{pmatrix}$ nazivamo *generalisana ađungovana matrica za A reda p.*

Izračunavanje matričnih karakteristika

Sledeće procedure služe za detekciju matrica posebnog tipa.

```
SquareMatrixQ[a_]:=Length[a]==Length[a[[1]]] /; MatrixQ[a]
OrtogonalMatrixQ[a_]:=
  a.Transpose[a]==IdentityMatrix[Length[a]] &&
  Transpose[a].a==IdentityMatrix[Length[a]] /; SquareMatrixQ[a]
HermitianMatrixQ[a_]:= a==Transpose[a] /;
           SquareMatrixQ[a]
```

Rank date matrice A jednak je sumi nenula elemenata koji leže na glavnoj dijagonali u redukovanoj echelonskoj formi matrice A.

```
Rank[a_]:=
  Block[{b=a, c, m,n, i},
        c=RowReduce[b]
        {m,n} =Dimensions[c]
        Sum[c[[i,i]], {i,Min[m,n]}]
      ] /; MatrixQ[a]
```

Indeks kvadratne matrice A je prvi nenula ceo broj k koji ispunjava $\mathrm{rank}(A^{k+1}) = \mathrm{rank}(A^k)$.

```
Index[a_]:=
  Block[{b=a,c=IdentityMatrix[Length[a]],d=a,k=0},
        While[Rank[c]=!=Rank[d],
                     d=d.b; c=c.b; k+=1
        ];
        k
  ] /; SquareMatrixQ[a]
```

Izračunavanje pravougaonih determinanti

```
DetS[a_]:=
  Block[{b=a, f, s, m, k, l},
        m=Simplify[Minors[b, Rank[b]]];
        f=Length[m]; s=Length[First[m]];
        Sum[m[[k,l]], {k,f}, {l,s}]
      ]/; MatrixQ[a]
```

Generalisana determinanta reda t

$$\sum_{\substack{1\le\delta_1<\ldots<\delta_r\le n\\1\le\gamma_1<\ldots<\gamma_r\le m}} \overline{R}\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix} A\begin{pmatrix}\gamma_1 & \ldots & \gamma_r\\ \delta_1 & \ldots & \delta_r\end{pmatrix}.$$

može da se izračuna pomoću procedure *GDetR*. Formalni parametri a i r označavaju matrice A i R, respektivno, dok parametar t predstavlja veličinu selektovanih minora.

```
GDetR[a_,r_, t_Integer]:=
  Block[{b=a, ra=r, f, s, k, l, ma,mc},
        ma=Simplify[Minors[b,t]];
        mc=Simplify[Minors[ra,t]];
        {f,s}=Dimensions[ma];
        Sum[mc[[k,l]] ma[[k,l]], {k,f}, {l,s}]
      ]/; MatrixQ[a] && MatrixQ[r] &&
         Dimensions[a]==Dimensions[r] && Rank[a]==Rank[r]
```

U cilju implementacije opšte determinantske reprezentacije, potrebne su dve korisne funkcije. Prva od tih funkcija generiše podmatricu date matrice A, koja se dobija odbacivanjem i-te vrste i j-te kolone u A.

```
MatrixComp[a_, i_Integer, j_Integer]:=
  Block[{b=a},
        b=Drop[b,{i,i}]
        b=Transpose[Drop[Transpose[b],{j,j}]];
      ]/; MatrixQ[a]
```

U drugoj funkciji se generiše podmatrica od A determinisana vrstama $p_1,\ldots,p_t$ i kolonama $q_1,\ldots,q_t$.

```
Minor[a_, p_List, q_List, t_Integer]:=
  Block[{b=a,i,j, c},
        c=IdentityMatrix[r];
        For[i=1, i<=t, i++,
              For[j=1, j<=t, j++,
                       c[[i,j]]=b[[p[[i]],q[[j]]]]
           ]   ];
        c
      ]/; Simplify[MatrixQ[a]]
```

Sada je opisana implementacija *opšte derminantske reprezentacije reda* $t\le r$ za $A\in\mathbb{C}_r^{m\times n}$:

$$a_{ij}^{(\dagger,R,t)}=\frac{\sum\limits_{(\alpha,\beta)\in\mathcal{N}(j,i,t)}\left[\ \overline{R}_{\beta}^{\alpha}\ \right]\frac{\partial}{\partial a_{ji}}\left[\ A_{\alpha}^{\beta}\ \right]}{\mathrm{DET}_{(\mathrm{R},\mathrm{t})}(\mathrm{A})},\qquad\left(\begin{array}{c}1\le i\le n\\1\le j\le m\end{array}\right).$$

Napomenimo da algoritam za generisanje kombinacija reda t skupa $\{1,\ldots,n\}$ [90] može da se napiše u sledećem kodu:

```
While[j>=1,
         If[j>=1,
                 For[i=t, i>=j, i--,
                       p[[i]]=p[[j]]+i-j+1; p1[[i]]=p[[i]]
      ]     ]      ];
```

Konačno, u proceduri $Rinverse[,]$ je implementirana opšta determinantska reprezentacija. Formalni parameteri a i r predstavljaju ulazne matrice A i R, respektivno.

```
RInverse[a_,r_]:=
Block[{b=a,ra=r,t,p,q, m,n, w,v, i,j,k, j1, p1,q1,
           pr,pr1, awv,mr,mrr, mc,s,inv, sw,am},
  inv=Transpose[b];
  t=Rank[b]; {m,n}=Dimensions[b];
  d=GDetR[b,ra,t];
  While[d==0,     d=GDetR[b,ra,t];     t-- ];
  p=q=Range[t];     p1=q1=q;
  For[v=1, v<=n, v++,
   For[w=1, w<=m, w++,
       s=0;
       If[t==m, j=1, j=m];
       While[j>=1,
            If[t==n, j1=1, j1=n];
            While[j1>=1,
                pr=pr1=1;
                While[pr<=t && p[[pr]]=!=w, pr++];
                While[pr1<=t && q[[pr1]]=!=v, pr1++];
                If[pr<=t && pr1<=t,
                     mr=Minor[b,p,q,t];
                     mrr=Minor[ra,p,q,t];
                     mc=Det[mrr];
                     am=Det[MatrixComp[mr,pr,pr1]];
                     awv=(-1)^(pr+pr1) am mc,
                     awv=0
```

```
                    ];
                    s+=awv;
                    If[q[[t]]==n, j1--, j1=t ];
                    If[j1>=1,
                         For[i=t, i>=j1, i--,
                            q[[i]]=q[[j1]]+i-j1+1;
                            q1[[i]]=q[[i]]
                    ]    ]
                ];
                q1=q=Range[t];
                If[p[[t]]==m, j--, j=t ];
                If[j>=1,
                    For[i=t, i>=j, i--,
                          p[[i]]=p[[j]]+i-j+1; p1[[i]]=p[[i]]
                ]    ]
           ];
           inv[[v,w]]=s/d
           p=q=Ranke[t];     p1=q1=q
      ]
     ];
     Simplify[inv]
   ]/; MatrixQ[a]
```

Pomoću opisane funkcije *RInverse* mogu se definisati funkcije za izračunavanje svih refleksivnih g-inverza. Posebno, funkcija za izračunavanje Moore-Penroseovog inverza za $A \in \mathbb{C}_r^{m\times n}$ može da se definiše pomoću

```
MPInverse[a_]:= RInverse[a_,a_]
```

Takođe, grupni inverz kvadratne matrice može da se izračuna na sledeći način:

```
GroupInverse[a_]:=
  Block[{b=a},
            If[Index[a] == 1,
                    RInverse[b,Transpose[b]],
                    Print["Group inverse does not exists"]
            ]
  ]/; SquareMatrixQ[a]
```

Napomena 2.2 *Izračunavanje generalisanih inverza pomoću opšte determinantske reprezentacije je direktni metod, i ne koristi Gaussovu eliminaciju.*

Primer 2.15 Uočimo test-matricu S_5 iz [86], za slučaj $a = 1$, tj.

$$S_5 = \begin{bmatrix} 2 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 1 & 2 \end{bmatrix}.$$

Njena full-rang faktorizacija je, na primer:

$$P = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Za matrice W_1 i W_2 mogu se uzeti

$$W_1 = \begin{bmatrix} 1 & 2 & 5 & 3 \\ -2 & 4 & 0 & 3 \\ 2 & 1 & 0 & -2 \\ 0 & 5 & 0 & 1 \\ 7 & 2 & -3 & 2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 2 & -2 & 1 & 1 & -5 \\ 3 & 0 & 1 & 4 & 0 \\ 0 & 2 & 1 & 3 & 4 \\ 7 & 1 & 1 & 9 & -3 \end{bmatrix}.$$

$RInverse[S_5,\ Transpose\ [W_1.W_2]]$ daje

$$S_5^{(1,2)} = \begin{bmatrix} \frac{28759}{10220} & -\frac{113}{140} & -\frac{1}{28} & \frac{151}{20} & -\frac{61317}{10220} \\ \frac{472}{73} & -2 & 1 & -1 & -\frac{399}{73} \\ -\frac{300}{73} & 1 & 0 & 1 & \frac{227}{73} \\ \frac{186}{73} & -1 & 1 & -2 & -\frac{113}{73} \\ -\frac{46539}{10220} & \frac{253}{140} & -\frac{27}{28} & -\frac{131}{0} & \frac{79097}{10220} \end{bmatrix}.$$

$RInverse[S_5,\ P.Transpose\ [W_1]]$ daje

$$S_5^{(1,2,3)} = \begin{bmatrix} \frac{223}{140} & -\frac{113}{140} & -\frac{1}{28} & \frac{151}{20} & -\frac{123}{140} \\ \frac{1}{2} & -2 & 1 & -1 & \frac{1}{2} \\ -\frac{1}{2} & 1 & 0 & 1 & -\frac{1}{2} \\ \frac{1}{2} & -1 & 1 & -2 & -\frac{1}{2} \\ -\frac{223}{140} & \frac{253}{140} & -\frac{27}{28} & -\frac{131}{20} & \frac{223}{140} \end{bmatrix}.$$

$RInverse[S_5,\ Transpose\ [W_2].Q]$ daje

$$S_5^{(1,2,4)} = \begin{bmatrix} -\frac{127}{146} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{127}{146} \\ \frac{472}{73} & -2 & 1 & -1 & -\frac{399}{73} \\ -\frac{300}{73} & 1 & 0 & 1 & \frac{227}{73} \\ \frac{186}{73} & -1 & 1 & -2 & -\frac{113}{73} \\ -\frac{127}{146} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{127}{14} \end{bmatrix}.$$

Vrednost izraza $RInverse[\ S_5,\ S_5\ ]$ je

$$S_5^{\dagger} = \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & -2 & 1 & -1 & \frac{1}{2} \\ -\frac{1}{2} & 1 & 0 & 1 & -\frac{1}{2} \\ \frac{1}{2} & -1 & 1 & -2 & \frac{1}{2} \\ 0 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix},$$

što se može dobiti iz [85].

Primer 2.16 Potpuna rang faktorizacija za $A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}$ je određena sledećim matricama:

$$P = \begin{bmatrix} 2 & -1 \\ -1 & -1 \\ 3 & 0 \\ -3 & 0 \\ 1 & 1 \\ -2 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & -\frac{1}{3} & \frac{1}{3} & 1 \\ 1 & -\frac{2}{3} & -\frac{1}{3} & 0 \end{bmatrix}.$$

Primenom funkcje $RInverse$, za

$$W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 4 & 2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & -1 \\ 0 & -1 & 0 & 0 & -2 & 1 \end{bmatrix},$$

dobija se:

$RInverse[A,\ Transpose\ [W_1.W_2]]$ gives

$$A^{(1,2)} = \begin{bmatrix} -\frac{10}{17} & \frac{44}{51} & -\frac{10}{17} & -\frac{10}{51} & \frac{88}{51} & -\frac{44}{51} \\ \frac{13}{17} & -\frac{47}{51} & \frac{13}{17} & \frac{13}{51} & -\frac{94}{51} & \frac{47}{51} \\ -\frac{5}{17} & \frac{22}{51} & -\frac{5}{17} & -\frac{5}{51} & \frac{44}{51} & -\frac{22}{51} \\ \frac{6}{17} & -\frac{2}{17} & \frac{6}{17} & \frac{2}{17} & -\frac{4}{17} & \frac{2}{17} \end{bmatrix},$$

$RInverse[A,\ \ P.Transpose\ [W_1]]$ daje

$$A^{(1,2,3)} = \begin{bmatrix} -\frac{2}{17} & -\frac{3}{17} & \frac{1}{17} & -\frac{1}{17} & \frac{3}{17} & \frac{2}{17} \\ \frac{19}{102} & \frac{10}{51} & -\frac{1}{102} & \frac{1}{102} & -\frac{10}{51} & -\frac{19}{102} \\ -\frac{1}{17} & -\frac{3}{34} & \frac{1}{34} & -\frac{1}{34} & \frac{3}{34} & \frac{1}{17} \\ \frac{7}{51} & \frac{2}{51} & \frac{5}{51} & -\frac{5}{51} & -\frac{2}{51} & -\frac{7}{51} \end{bmatrix},$$

$RInverse[A,\ \ Transpose\ [W_2]].Q$ daje

$$A^{(1,2,4)} = \begin{bmatrix} -\frac{11}{17} & \frac{43}{51} & -\frac{11}{17} & -\frac{11}{51} & \frac{86}{51} & -\frac{43}{51} \\ \frac{7}{17} & -\frac{32}{51} & \frac{7}{17} & \frac{7}{51} & -\frac{64}{51} & \frac{32}{51} \\ \frac{4}{17} & -\frac{11}{51} & \frac{4}{17} & \frac{4}{51} & -\frac{22}{51} & \frac{11}{51} \\ \frac{1}{17} & \frac{10}{51} & \frac{1}{17} & \frac{1}{51} & \frac{20}{51} & -\frac{10}{51} \end{bmatrix},$$

$MPInverse[A]$ proizvodi

$$A^{\dagger} = \begin{bmatrix} -\frac{5}{34} & -\frac{3}{17} & \frac{1}{34} & -\frac{1}{34} & \frac{3}{17} & \frac{5}{34} \\ \frac{4}{51} & \frac{13}{102} & -\frac{5}{102} & \frac{5}{102} & -\frac{13}{102} & -\frac{4}{51} \\ \frac{7}{102} & \frac{5}{102} & \frac{1}{51} & -\frac{1}{51} & -\frac{5}{102} & -\frac{7}{102} \\ \frac{1}{17} & -\frac{1}{34} & \frac{3}{34} & -\frac{3}{34} & \frac{1}{34} & -\frac{1}{17} \end{bmatrix}.$$

Primer 2.17 Uočimo test-matricu A iz [85]

$$A = \begin{bmatrix} 1+x & x & x+1 \\ x & x-1 & x \\ x+1 & x & x+1 \end{bmatrix}.$$

In[1]:=MPInverse[A]

$$\text{Out[1]}=A^{\dagger} = \begin{bmatrix} \frac{1-x}{4} & \frac{x}{2} & \frac{1-x}{4} \\ \frac{x}{2} & -1-x & \frac{x}{2} \\ \frac{1-x}{4} & \frac{x}{2} & \frac{1-x}{4} \end{bmatrix}.$$

Primer 2.18 Test matrica B data sa

$$B = \begin{bmatrix} x+5 & x+3 & x+2 & x+4 & x+3 & x+2 \\ x+3 & x+4 & x+2 & x+3 & x+3 & x+2 \\ x+2 & x+2 & x+2 & x+2 & x+2 & x+1 \\ x+4 & x+3 & x+2 & x+3 & x+3 & x+2 \\ x+3 & x+3 & x+2 & x+3 & x+2 & x+2 \\ x+2 & x+2 & x+1 & x+2 & x+2 & x \\ x+1 & x & x+1 & x+1 & x+1 & x-1 \end{bmatrix}.$$

$MPInverse[B]$ daje rezultat

$$B^{\dagger} = \begin{bmatrix} -3-x & -\frac{9+3x}{4} & -\frac{11+5x}{4} & 4+x & 3+x & \frac{5+3x}{4} & \frac{3+x}{4} \\ -3-x & -\frac{6+2x}{4} & -\frac{10+5x}{4} & 3+x & 3+x & \frac{6+3x}{4} & \frac{2+x}{4} \\ -2-x & -\frac{5+3x}{4} & -\frac{3+5x}{4} & 2+x & 2+x & \frac{1+3x}{4} & \frac{3+x}{4} \\ 4+x & \frac{9+3x}{4} & \frac{11+5x}{4} & -5-x & -3-x & -\frac{5+3x}{4} & -\frac{3+x}{4} \\ 3+x & \frac{9+3x}{4} & \frac{11+5x}{4} & -3-x & -4-x & -\frac{5+3x}{4} & -\frac{3+x}{4} \\ 2+x & \frac{5+3x}{4} & \frac{7+5x}{4} & -2-x & -2-x & -\frac{5+3x}{4} & -\frac{3+x}{4} \end{bmatrix}.$$

2.1.6 Leverrier-Faddeev metod

Jedan od iterativnihnih metoda za izračunavanje Moore-Penroseovog inverza potiče od Decella [14], i zasniva se na modifikovanom Leverrier metodu za određivanje koeficijenata karakterističnog polinoma matrice AA^*. Metod proizilazi iz sledeće teoreme, koja predstavlja uopštenje Frameovog rezultata.

Teorema 2.18 *Za zadatu matricu $A \in \mathbb{C}^{m\times n}$ defiinisati nizove*

$$B_j, \quad j = 0, 1, \dots, \qquad p_j, \quad j = 0, 1, \dots$$

pomoću

$$B_0 = I_m, \quad p_j = \frac{1}{j} Tr(AA^* B_{j-1}), \quad B_j = AA^* B_{j-1} - p_j I_m.$$

Tada je

$$A^{\dagger} = \frac{1}{p_k} A^* B_{k-1},$$

gde je $k \neq 0$ najveći prirodan broj za koji je $p_k \neq 0$. Ako je $p_1 = 0$, tada je $A^{\dagger} = \mathbb{O}$.

Algoritam 2.2 (Mpd1) *Za datu matricu A.*

Korak 1. Izračunati $C = AA^*$ i postaviti $B_0 = I_m$, $j = 0$.

Korak 2. Postaviti $j = j + 1$.

Korak 3. Izračunati

$$D = CB_{j-1}, \quad p_j = j^{-1}\mathrm{Tr(D)}, \quad \mathrm{B_j = D - P_j I_m}.$$

Korak 4. Ponavljati Korak 2. i Korak 3., sve dok se ne ispuni uslov $p_j = 0$.

Korak 5. Ako je $j = 1$ *i* $p_1 = 0$, *tada je* $A^\dagger = \mathbb{O}$, *a inače je* $A^\dagger = p_{j-1}^{-1} A^* B_{j-2}$.

Algoritam se može implementirati na sledeći način:

```
Mpd1[t_]:=
   Block[{a=t,bj,bj1,bj2,act,amp,e,c,d,j=0,m,n,pj=1},
     act=a.Conjugate[Transpose[a]];
     {m,n}=Dimensions[a]; e=IdentityMatrix[m];
     c=a.act;
     bj=bj1=e;
     While[pj!=0,
           j++;  d=c.bj;  pj1=pj;
           pj=j^(-1) Sum[d[[i,i]],{i,m}];
           bj2=bj1;bj1=bj; bj=d-pj e;
     ];
     If[j==1,
        amp=Table[{m},{n}],
        amp=pj1^(-1) act.bj2
     ];
     Return[amp];
   ]
```

Postoji analogni metod za izračunavanje Drazinovog inverza. Ovaj metod koristi da se Drazinov inverz može predstaviti kao matrični polinom po stepenima polazne matrice. Jednu od tih metoda je izložio Grevile u [33], dokazujući da je za izračunavanje Drazinovog inverza dovoljno odrediti koeficijente karakterističnog polinoma polazne matrice. U tom metodu, Grevile je koristio metod za izračunavanje koeficijenata karakterističnog polinoma koji je još 1840. godine dao Leverrier, a koji su modifikovali Souriau i Frame.

Teorema 2.19 *Ako je za zadatu matricu* $A \in \mathbb{C}^{n\times n}$,

$$\begin{aligned} B_0 &= I_n, \\ p_j &= j^{-1} Tr(AB_{j-1}), \quad B_j = AB_{j-1} - p_j I_n, \quad j = 1, 2, \ldots, \end{aligned}$$

tada je

$$A_d = P_s^{-k-1} A^k B_{s-1}^{k+1},$$

gde je s *najveći prirodan broj za koji je* $p_s \neq 0$, $k = r - s$, r *najmanji prirodan broj za koji je* $B_r = \mathbb{O}$.

U slučaju nesingularne matrice A, primenom poslednjeg metoda se dobija $k = 0$, $s = n$ i $A_d = A^{-1}$, te Drazinov inverz postaje $A^{-1} = p_n^{-1} B_{n-1}$, što predstavlja poznati rezultat koji su izveli Souriau i Frame.

```
Drzinv[t_]:=
   Block[{a=t,bj,bj1,bs1,drz,nul,e,c,j=0,m,n,pj=1,
         k,r,s,ps,nadjenr,nadjens},
     nadjenr=nadjens=True;
     {m,n}=Dimensions[a];
     If[m!=n, Print["Matrica nije kvadratna"],
        e=IdentityMatrix[m];
        bj1=bj=bs1=e;
        While[(nadjens || nadjenr),
           j++;  c=a.bj;
           pj=Simplify[j^(-1)*Sum[c[[i,i]],{i,m}]];
           bj1=bj; bj=Simplify[a.bj1-pj*e];
           {m,n}=Dimensions[bj];  nul=Table[0,{m},{n}];
           If[pj!=0, s=j; ps=pj; bs1=bj1, nadjens=False];
           If[bj==nul, r=j; nadjenr=False ]
        ];
        k=r-s;
        drz=Simplify[ps^(-k-1)*MatrixPower[a,k].MatrixPower[bs1,k+1]];
        Return[drz];
   ] ]
```

Primer 2.19 Za test matricu iz [85] primenom funkcije $Drzinv[\,]$ i $Mpd1[\,]$ dobijamo:

In[2]:=Drzinv[{{1+x,x,1+x},{x,x-1,x},{1+x,x,1+x}]

pj=1+3 x

bj={{-2 x,x,1+x},{x,-2 (1+x),x},{1+x,x,-2 x}}

pj=2

bj={{-1,0,1},{0,0,0},{1,0,-1}}

pj=0

bj={{0,0,0},{0,0,0},{0,0,0}}

Out[2]= $\begin{bmatrix} \frac{1-x}{4} & \frac{x}{2} & \frac{1-x}{4} \\ \frac{x}{2} & -1-x & \frac{x}{2} \\ \frac{1-x}{4} & \frac{x}{2} & \frac{1-x}{4} \end{bmatrix}$

In[3]:=Mpd1[{{1+x,x,1+x},{x,x-1,x},{1+x,x,1+x}]

pj=$7 + 27x + 27x^2 + 27x^3$

bj=$\{\{-3(1 + 5x + 4x^2 + 6x^3), 3x(1 + 2x + 3x^2), 4 + 12x + 15x^2 + 9x^3\},$
$\{3x(1 + 2x + 3x^2), -2(4 + 12x + 15x^2 + 9x^3), 3x(1 + 2x + 3x^2)\},$
$\{4 + 12x + 15x^2 + 9x^3, 3x(1 + 2x + 3x^2), -3(1 + 5x + 4x^2 + 6x^3)\}\}$

pj=8

bj={{-4,0,4},{0,0,0},{4,0,-4}}

pj=0
bj={{0,0,0},{0,0,0},{0,0,0}}

Out[3]= $\begin{bmatrix} \frac{1-x}{4} & \frac{x}{2} & \frac{1-x}{4} \\ \frac{x}{2} & -1-x & \frac{x}{2} \\ \frac{1-x}{4} & \frac{x}{2} & \frac{1-x}{4} \end{bmatrix}$

2.1.7 Generalisani inverzi i granični procesi

U [91, 90] je uvedena iterativna metoda za implementaciju limit reprezentacije Moore-Penrose-ovog inverza koja je ustanovljena u [5, 6]. U [71] je opisan sledeći algoritam koji generalizuje taj metod.

Lema 2.1 [71] *Neka su date dve proizvoljne kompleksne matrice R i S dimenzije $p \times q$, i kompleksni vektor y dimenzije $p \times 1$. Predpostavimo da su elementi vektora y označeni sa y_t, $t = 1, \ldots, p$. Neka su vrste matrice R i S i jedinična matrica I_p označene sa r_t, s_t i i_t, $t = 1, \ldots, p$ respektivno. Sledeći iterativni metod*

$$\begin{aligned}
\gamma_{t+1}^{\alpha} &= \gamma_t^{\alpha} - \frac{\gamma_t^{\alpha} s_{t+1}^* r_{t+1} \gamma_t^{\alpha}}{\alpha + s_{t+1} \gamma_t^{\alpha} r_{t+1}^*}, \quad \gamma_0^{\alpha} = I_q, \quad \alpha > 0, \\
x_{t+1}^{\alpha} &= x_t^{\alpha} + \frac{\gamma_t^{\alpha} r_{t+1}^*}{\alpha + s_{t+1} \gamma_t^{\alpha} r_{t+1}^*} \left(y_{t+1} - s_{t+1} x_t^{\alpha} \right), \quad x_0^{\alpha} = \vec{0} \\
X_{t+1}^{\alpha} &= X_t^{\alpha} + \frac{\gamma_t^{\alpha} r_{t+1}^*}{\alpha + s_{t+1} \gamma_t^{\alpha} r_{t+1}^*} \left(i_{t+1} - s_{t+1} X_t^{\alpha} \right), \quad X_0^{\alpha} = \mathbb{O}, \\
& t = 0, \ldots, p-1
\end{aligned}$$

proizvodi sledeću matricu:

$$X_p^{\alpha} = (\alpha I_q + R^* S)^{-1} R^*, \ \ x_p^{\alpha} = (\alpha I_q + R^* S)^{-1} R^* y, \ \ \gamma_p^{\alpha} = (\alpha I_q + R^* S)^{-1} \alpha.$$

U [71] je data sledeća šema za izračunavanje, koja obezbeđuje indirektno smanjenje vrednosti α.

Algoritam 2.3 *Iterativni proces zadovoljava*

$$X_{pN}^{\alpha} = X_p^{\alpha/N}, \quad \gamma_{pN}^{\alpha} = \gamma_p^{\alpha/N}, \quad N = 1, 2, \ldots$$

gde su

$$X_{pN}^{\alpha} = (\alpha I_q + R_{pN}{}^* S_{pN})^{-1} R_{pN}{}^*, \ \ \gamma_{pN}^{\alpha} = (\alpha I_q + R_{pN}{}^* S_{pN})^{-1} \alpha$$

i $R_{pN}{}^*$, s_{pN} sledeće blok matrice

$$R_{pN}{}^* = R_N{}^* = \begin{bmatrix} R^* & \ldots & R^* \end{bmatrix}, \qquad S_{pN} = S_N = \begin{bmatrix} S \\ \cdots \\ S \end{bmatrix}.$$

Glavni problem u implementaciji za limit representaciju obuhvata povećanje reda blok matrice R_N i S_N. Da bi se izbegao ovaj problem, u [71] je predložena MATHEMATICA funkcija `Mod`.

Ulazni parametri algoritma su:

$r\_, s\_$: ulazne matrice R i S;

$it\_$: propisani broj iteracija;

$alpha\_$: početna vrednost parametra α.

```
{p,q}=Dimensions[r];
in=IdentityMatrix[q]; im=IdentityMatrix[p];
g0=in; x0=in-in; i=1;
While[i<=it,j=i;
   If[i>p,j=Mod[i,p];If[j==0,j=p] ];
   g1=g0-(g0.Transpose[{r[[j]]}].{s[[j]]}.g0)/
      (alpha+({s[[j]]}.g0.Transpose[{r[[j]]}])[[1,1]]);
   x1=x0+g0.Transpose[{r[[j]]}].(im[[j]]-{s[[j]]}.x0)/
      (alpha+({s[[j]]}.g0.Transpose[{r[[j]]}])[[1,1]]);
   g0=g1; x0=x1; i=i+1];
Return[x1];
```

S'obzirom da je konvergencija limit algoritma veoma siromašna [85], data je implementacija algoritma u delovima određenim brojem iteracija. Kriterijum zaustavljanja algoritma je $\|X_{t+1} - X_t\| \leq \epsilon$ gde je ϵ najmanji realan broj, X_{t+1} i X_t su dve uspešne iteracije i $\|X\|$ je izabrana norma matrice. Na primer, za $X = (x_{ij})$ koristi se norma $\|X\| = \sqrt{\sum x_{ij}^2}$, koja je implementirana u sledećoj funkciji:

```
norm[a_]:=Block[{m,n,i,j,s},
      {m,n}=Dimensions[a];
      s=Sum[m[[i,j]]^2,{i,m},{j,n}];
      Return[N[Sqrt[s],20]]]
```

Poboljšanje implementacije je bazirano na sledećem rezultatu.

Teorema 2.20 *Neka su date kompleksne matrice R i S dimenzije $p \times q$. Ako je b proizvoljno određeni broj iteracija, $c = Quotient[b,p]$, $d = Mod[b,p]$, onda iterativna sekvenca definisana u prethodnom daje sledeće vrednosti:*

$$\begin{aligned} \gamma_{p+d}^{\alpha/c} &= \gamma_b^\alpha = \left(\frac{\alpha}{c} I_q + R_{\underline{d}}{}^* S_{\underline{d}}\right)^{-1} \frac{\alpha}{c} \\ X_{p+d}^{\alpha/c} &= X_b^\alpha = \left(\frac{\alpha}{c} I_n + R_{\underline{d}}{}^* S_{\underline{d}}\right)^{-1} R_{\underline{d}}{}^* I_{\underline{d}}, \end{aligned}$$

gde su $R_{\underline{d}}$ i $S_{\underline{d}}$, $d \times q$ submatrice iz R i S koje sadrže prvih d vrsta u R i S, respektivno, a $R_{\underline{d}}{}^*$ je $n \times d$ matrica, jednaka konjugovano transponovanoj submatrici $R_{\underline{d}}$.

Dokaz. Saglasno opisanom algoritmu, posle prvih p iteracija dobija se:

$$\gamma_p^{\alpha/c} = \gamma_{pc}^{\alpha}, \quad X_p^{\alpha/c} = X_{pc}^{\alpha}.$$

Preostalim koracima d iteracije dobijaja se

$$\gamma_{p+d}^{\alpha/c} = \gamma_{pc+d}^{\alpha} = \gamma_b^{\alpha}, \quad X_{p+d}^{\alpha/c} = X_{pc+d}^{\alpha} = X_b^{\alpha}.$$

Ovim je dokaz završen. □

Iterativni korak *Korak 2* može se poboljšati na sledeći način.

```
c=Quotient[it,m]; d=Mod[it,m]; alpha=alpha/c;
While[norm[x1-x0]>=eps, i=1;
   While[i<=p+d, j=i;
     If[i>m,j=Mod[i,m];If[j==0,j=m];
     g1=g0-(g0.Transpose[{s[[j]]}].{s[[j]]}.g0)/
        (alpha+({s[[j]]}.g0.Transpose[{s[[j]]}])[[1,1]]);
     x1=x0+g0.Transpose[{s[[j]]}].(im[[j]]-{s[[j]]}.x0)/
        (alpha+({s[[j]]}.g0.Transpose[{s[[j]]}])[[1,1]]);
     g0=g1; x0=x1; i=i+1
]  ];
```

2.1.8 Konačni algoritam za inverze racionalnih matrica

Sledećom lemom predstavljena je potpuna rang reprezentacija za različite slučajeve generalisanih inverza date racionalne matrice. Ova poznata reprezentacija generalizuje rezultat iz [55].

Lema 2.2 *Neka je $A(s) \in \mathbb{C}(s)_r^{m\times n}$ proizvoljna racionalna matrica dimenzije $m \times n$ ranga r i neka je $A(s) = P(s)Q(s)$ potpuna rang faktorizacija za $A(s)$. Sledi karakterizacija za različite slučajeve generalisanih inverza za $A(s)$:*

(a) $A(s)\{2\} = \left\{G(s)(H(s)A(s)G(s))^{-1}G(s) :\ H(s) \in \mathbb{C}(s)_t^{t\times m},\ G(s) \in C(s)_t^{n\times t},\ t \le r\right\}$.

(b) $A(s)\{1,2\} = \left\{G(s)(H(s)A(s)G(s))^{-1}H(s) :\ H(s) \in \mathbb{C}(s)_r^{r\times m},\ G(s) \in \mathbb{C}(s)_r^{n\times r}\right\}$.

(c) $A(s)\{1,2,3\} = \left\{G(s)(P(s)^*A(s)G(s))^{-1}P(s)^* :\ G(s) \in \mathbb{C}(s)_r^{n\times r}\right\}$.

(d) $A(s)\{1,2,4\} = \left\{Q(s)^*(H(s)A(s)Q(s)^*)^{-1}H(s) :\ H(s) \in \mathbb{C}(s)_r^{r\times m}\right\}$.

(e) $A(s)^\dagger = Q(s)^*(P(s)^*A(s)Q(s)^*)^{-1}P(s)^*$.

Koristeći Lemu 2.2, u sledećoj teoremi predstavljen je generalni konačni algoritam za izračunavanje različitih klasa generalilisanih inverza za $A(s)$.

Teorema 2.21 *Neka su $R(s)$, $T(s)$ i $A(s)$ izabrane racionalne matrice dimenzije $m \times n$ i $e \geq 1$ proizvoljni ceo broj. Pretpostavimo da je $A(s) = P(s)Q(s)$ potpuna rang faktorizacija za $A(s)$. Uočimo racionalne $a_j(s)$, $a'_j(s)$ i racionalne matrice $B_j(s)$, $B'_j(s)$ $j = 0, 1, \ldots, m$ koje su definisane sledećim rekurzivnim algoritmom:*

$$\begin{aligned} B_0(s) &= I_m, \quad a_0(s) = 1 && (2.1.1)\\ A_j(s) &= T(s)R(s)^* B_{j-1}(s), && (2.1.2)\\ a_j(s) &= -\frac{Tr(A_j(s))}{j}, \quad B_j(s) = A_j(s) + a_j(s)I_q, \; j = 1, \ldots, m, && (2.1.3)\end{aligned}$$

$$\begin{aligned} B'_0(s) &= I_n, \quad a_0(s) = 1 && (2.1.4)\\ A'_j(s) &= R(s)^* T(s) B'_{j-1}(s), && (2.1.5)\\ a'_j(s) &= -\frac{Tr(A'_j(s))}{j}, \quad B'_j(s) = A'_j(s) + a'_j(s)I_q, \; j = 1, \ldots, n. && (2.1.6)\end{aligned}$$

Neka je k najveći ceo broj takav da je $a_k(s) \neq 0$ i k' najveći ceo koji zadovoljava $a'_{k'}(s) \neq 0$. Posmatrajmo matrice

$$X_e(s) = \begin{cases} (-1)^e \frac{R(s)^* B_{k-1}(s)^e}{a_k(s)^e}, & k \geq 1 \\ \mathbb{O}, & k = 0 \end{cases} \tag{2.1.7}$$

and

$$X'_e(s) = \begin{cases} (-1)^e \frac{B'_{k'-1}(s)^e R(s)^*}{a'_{k'}(s)^e}, & k' \geq 1 \\ \mathbb{O}, & k' = 0. \end{cases} \tag{2.1.8}$$

U slučaju $e = 1$ imamo

$$X_1(s) = X'_1(s). \tag{2.1.9}$$

Takođe, su ispravna sledeća tvrđenja:

(i) U slučaju $e = 1$, $R(s) = T(s) = A(s)$ imamo $X_1(s) = A(s)^\dagger$.

(ii) Ako je $A(s)$ racionalna matrica dimenzije $n \times n$, izborom vrednosti $e = 1$, $m = n$, $R(s)^* = A(s)^l$, $l \geq \text{ind}(A)$, $T(s) = A(s)$, dobijamo $X_1(s) = A(s)^D$.

(iii) Ako je $T(s) = A(s)$, $n > m = \text{rank}(A(s))$, za proizvoljnu $R(s) \in \mathbb{C}(s)^{m\times n}_q$, takvu da je $A(s)R^*(s)$ invertibilna, u slučaju $e = 1$ dobijamo $X_1(s) = A(s)^{-1}_R$.

(iv) Ako je $A(s) \in \mathbb{C}(s)^{n\times n}$, $m = n$, $e = 1$, $R(s)^* = A(s)^k$, $T(s) = I_n$, onda $X_1(s)$ postoji, ako i samo ako $\text{ind}(A(s)) = k$, i $X_1(s) = A(s)A(s)^D$.

(v) Ako je $m = n$, $e = l + 1$, $T(s)R(s)^* = A(s)$, $R(s)^* = A(s)^l$, $l \geq \text{ind}(A(s))$, dobijamo $X_e(s) = X'_e(s) = A(s)^D$.

(vi) Za $A(s) \in \mathbb{C}(s)^{n\times n}_r$, u slučaju $e = 1$, $T(s) = R(s)^* = A(s)^l$, $l \geq \text{ind}(A(s))$, imamo $X_1(s) = (A(s)^D)^l$.

(vii) $X_1(s) \in A(s)\{2\} \Leftrightarrow e=1,\ T(s)=A(s),\ \ R(s)^*=G(s)H(s),$
$G(s)\in\mathbb{C}(s)^{n\times t},\ H(s)\in\mathbb{C}(s)^{t\times m},\ t\le r,\ \operatorname{rank}(H(s)A(s)G(s))=t.$

(viii) $X_1(s) \in A(s)\{1,2\} \Leftrightarrow e=1,\ T(s)=A(s),\ \ R(s)^*=G(s)H(s),$
$G(s)\in\mathbb{C}(s)^{n\times r},\ H(s)\in\mathbb{C}(s)^{r\times m},\ \operatorname{rank}(H(s)A(s)G(s))=r.$

(ix) $X_1(s) \in A(s)\{1,2,3\} \Leftrightarrow e=1,\ T(s)=A(s),\ \ R(s)^*=G(s)P(s)^*,$
$G(s)\in\mathbb{C}(s)^{n\times r},\ \operatorname{rank}(P(s)^*A(s)G(s))=r.$

(x) $X_1(s) \in A(s)\{1,2,4\} \Leftrightarrow e=1,\ T(s)=A(s),\ \ R(s)^*=Q(s)^*H(s),$
$G(s)\in\mathbb{C}(s)^{n\times r},\ H(s)\in\mathbb{C}(s)^{r\times m},\ \operatorname{rank}(H(s)A(s)Q(s)^*)=r.$

(xi) Ako je $T(s)=A(s)$, $m>n=\operatorname{rank}(A(s))$, za proizvoljnu $R(s)\in\mathbb{C}(s)_q^{m\times n}$, tako da je $R^*(s)A(s)$ invertibilna, u slučaju $e=1$ dobijamo $X_1(s)=A(s)_L^{-1}$.

Dokaz. Na početku dokazujemo da racionalna matrica $X_e(s)$ zadovoljava

$$X_e(s)=\lim_{\alpha\to 0} R(s)^*(\alpha I_m+T(s)R^*(s))^{-e}. \tag{2.1.10}$$

Na sličan način kao u [19] i [78], nije teško pokazati da je:

$$\begin{aligned}
R(s)^*(\alpha I_m+T(s)R(s)^*)^{-e} &= (-1)^e R(s)^*((-\alpha)I_m-T(s)R(s)^*)^{-e}\\
&=(-1)^e R(s)^*\frac{\big((-\alpha)^{m-1}+(-\alpha)^{m-2}B_1(s)+\cdots+(-\alpha)^{m-k}B_{k-1}(s)\big)^e}{((-\alpha)^m+a_1(s)(-\alpha)^{m-1}+\cdots+a_k(s)(-\alpha)^{m-k})^e}\\
&=(-1)^e R(s)^*\frac{\big((-\alpha)^{k-1}+(-\alpha)^{k-2}B_1(s)+\cdots+B_{k-1}(s)\big)^e}{((-\alpha)^k+a_1(s)(-\alpha)^{k-1}+\cdots+a_k(s))^e}.
\end{aligned}$$

Onad (2.1.10) sledi iz $\alpha\to 0$. Slično se može pokazati i sledeće:

$$X'_e(s)=\lim_{\alpha\to 0}(\alpha I_m+R^*(s)T(s))^{-e}R(s)^*. \tag{2.1.11}$$

Da bi dokazali (2.1.9), uvodimo sledeće tvrđenje:

$$\begin{aligned}
R(s)^*B_l(s) &= B'_l(s)R(s)^*\\
a_l &= a'_l, \qquad\qquad l\ge 0.
\end{aligned}$$

U slučaju $l=0$ tvrđenje se može direktno dokazati. Ako tvrđenje tačno za $l=j$, imamo

$$\begin{aligned}
a_{j+1}(s) &= -\frac{\operatorname{Tr}(T(s)R(s)^*B_j(s))}{j+1} = -\frac{\operatorname{Tr}(T(s)B'_j(s)R(s)^*)}{j+1}\\
&= -\frac{\operatorname{Tr}(R(s)^*T(s)B'_j(s))}{j+1} = -\frac{\operatorname{Tr}(A'_{j+1})}{j+1} = a'_{j+1}(s);
\end{aligned}$$

$$
\begin{aligned}
R(s)^* B_{j+1}(s) &= R(s)^*(T(s)R(s)^* B_j(s) + a_{j+1}(s)I_m) \\
&= (R(s)^* T(s) B'_j(s) R(s)^* + a'_{j+1}(s) R(s)^*) \\
&= (R(s)^* T(s) B'_j(s) + R(s)^* a'_{j+1}(s) I_n) R(s)^* = B'_{j+1}(s) R(s)^*.
\end{aligned}
$$

Kao posledica (2.1.9), (2.1.10) i (2.1.11) imamo

$$\lim_{\alpha\to 0} R^*(s)(\alpha I_m + T(s)R^*(s))^{-1} = \lim_{\alpha\to 0}(\alpha I_n + R^*(s)T(s))^{-1}R^*(s).$$

Sada se tvrđenje može završiti generalizovanjem poznatog rezultata o graničnim izrazima $\lim\limits_{\alpha\to 0} R^*(\alpha I_m + TR^*)^{-e}$, iz [62]. $\square$

Shodno rezultatima Teoreme 2.21 data su sledeća dva algoritma za izračunavanje različitih generalisanih inverza za $A(s)$.

Algoritam 2.4 *Predpostavimo da su $A(s), R(s), T(s) \in \mathbb{C}(s)^{m\times n}$ racionalne matrice.*

Korak 1. Konstruisati niz polinoma $\{a_0(s), \dots, a_k(s)\}$ i niz $\{B_0(s), \dots, B_{k-1}(s)\}$ polinomijalnih matrica kao u (2.1.1), (2.1.2) *i* (2.1.3), *gde k predstava $k = \max\{l : a_l(s) \neq 0\}$.*

Korak 2. Naći racionalnu matricu $X_e(s)$ kao u (2.1.7).

Algoritam 2.5 *Neka su date racionalne matrice $A(s), R(s), T(s) \in \mathbb{C}(s)^{m\times n}$.*

Korak 1. Konstruisati niz polinoma $\{a'_0(s), \dots, a'_{k'}(s)\}$ i niz $\{B'_0(s), \dots, B'_{k'-1}(s)\}$ polinomijalnih matrica kao u (2.1.4), (2.1.5) *i* (2.1.6), *gde k predstavlja $k' = \max\{l : a'_l(s) \neq 0\}$.*

Korak 2. Naći racionalne matrice $X'_e(s)$ kao u (2.1.8).

Napomena 2.3 *Konačni algoritam definisan u teoremi Teorem 2.21 predstavlja generalnu formu za sve poznate modifikacije Leverrier-Faddeevog algoritma za izračunavanje različitih generalisanih inverza racionalnih matrica, koje su razmatrane u* [22],[78] *i* [79].

U slučaju (i) Teoreme 2.21 dobijamo dobro poznati konačni algoritam za izračunavanje Moore-Penrose inverza racionalnih matrica, predstavljen u *[22]*.

U slučaju (ii) dobija se modifikovani Leverrier-Faddeev algoritam za izračunavanje Drazin inverza date racionalne matrice, razmatran u *[78]* i *[79]*.

Slično, u slučaju (v) predstavljena je generalizacija alternativne konačne reprezentacije Drazin inverza, prestavljen u *[19]*.

Algoritam 2.4 i Algoritam 2.5 predstavljaju spoljašnje inverze i g-inverze za $A(s)$. Šta više, racionalna matrica $X_1(s)$ je u generalnoj formi za neke specifične izraze uvedenih generalisanih inverza za $A(s)$, kao što su $A(s)A(s)^D$ ili $(A(s)^D)^l$.

Sada sledi implementacija *Algoritma 2.4* u paketu MATHEMATICA. Uglavnom se koriste sledeće mogućnosti paketa MATHEMATICA, verzija 4.1 (videti [83] i [84]) u simboličkom izračunavanju:

1. Simbolična matrična algebra, koja uključuje simboličke operacije nad vektorima i matricama.

2. Algebarske manipulacije, koje uključuju izračunavanja sa simbolima i operacijama nad algebarskim izrazima.

3. Ugrađena funkcija `Simplify[expr]`, koja algebarski izraz označen sa *expr*, vraća u pojednostavljenoj formi.

Sledećom funkcijom izračunava se matrica $X_e(s)$ definisana u (2.1.7). Napomenimo da je u sledećem kodu standardna funkcija Tr[$g\_$] koja daje trag matrice g.

```
GeneralRational1[r_, t_,e_]: =
Module[{Bi,Bi1,Bi2,inv,nul,im,Ai,ai=ai1=1,m,n,i=k=0,log=True,rct=Transpose[r]},
  {m,n}=Dimensions[r]; nul=Table[0,{m},{n}];
  im=IdentityMatrix[m]; Bi=Bi1=Bi2=im;
  While[log, i++; Ai=Simplify[t.rct.Bi];          (* Compute A[i] *)
      ai1=ai; ai=-Simplify[Tr[Ai]/i];             (* Compute a[i] *)
      Bi2=Bi1; Bi1=Bi; Bi=Simplify[Ai+ai*im];     (* Compute B[i] *)
      If[ai===0, k=i-1; log=False];
  ];
  If[k>0,inv = Simplify[(-1)^e*ai1^(-e)*rct.Bi2^e];
         Return[Simplify[inv]],                      (* k>0 *)
         Return[Transpose[nul]]                      (* k=0 *)
]  ]
```

Primer 2.20 Neka je data racionalna matrica iz [22].

$$A(s) = \begin{bmatrix} 1 & s+1 & s \\ \frac{1}{s} & \frac{3}{s+1} & \frac{1}{s+2} \end{bmatrix}.$$

U sličaju kad je $t = r = A(s)$ funkcija $GeneralRational1[r, t, 1]$ daje dobro poznat Moore-Penrose inverz iz [22]:

$$\begin{bmatrix} -\frac{s(12+s+6s^2+12s^3+4s^4)}{4-4s+7s^2-2s^3+17s^4+22s^5+6s^6} & \frac{s(1+s)^2(2+s)(2-s+2s^2+2s^3)}{4-4s+7s^2-2s^3+17s^4+22s^5+6s^6} \\ -\frac{(1+s)(-4-2s^2+s^3+s^4}{4-4s+7s^2-2s^3+17s^4+22s^5+6s^6} & \frac{s(1+s)(2+s)(-2+s-2s^2+3s^3+2s^4}{4-4s+7s^2-2s^3+17s^4+22s^5+6s^6} \\ \frac{s(2+s)(1+15s^2+7s^3)}{4-4s+7s^2-2s^3+17s^4+22s^5+6s^6} & -\frac{s^3(1+s)^2(2+s)(5+2s)}{4-4s+7s^2-2s^3+17s^4+22s^5+6s^6} \end{bmatrix}.$$

Primer 2.21 Za matricu

$$A(s) = \begin{bmatrix} s+1 & s & s+1 \\ s & s-1 & s \\ s+1 & s & s+1 \end{bmatrix}.$$

u slučaju kad je $r = A(s)^T$, $t = A(s)$ primenom funkcije $GeneralRational1[r, t, 1]$ dobijamo poznati Drazin inverz iz [75]:

$$A(s)^D = \begin{bmatrix} \frac{1-s}{4} & \frac{s}{2} & \frac{1-s}{4} \\ \frac{s}{2} & -1-s & \frac{s}{2} \\ \frac{1-s}{4} & \frac{s}{2} & \frac{1-s}{4} \end{bmatrix}.$$

Primer 2.22 Za matricu iz [20]:

$$A(s) = \begin{bmatrix} s^2+s & 1 & s^2+s \\ 1 & s^2-s & 1 \\ s^2+s & 1 & s^2+s \end{bmatrix}.$$

u slučaju kad je $T(s) = R(s) = A(s)$, dobijamo sledeći Moore-Penrose inverz za $A(s)$:

$$A(s)^\dagger \begin{bmatrix} \frac{(-1+s)s}{4(-1-s^2+s^4)} & \frac{1}{2+2s^2-2s^4} & \frac{(-1+s)s}{4(-1-s^2+s^4)} \\ \frac{1}{2+2s^2-2s^4} & \frac{s(1+s)}{-1-s^2+s^4} & \frac{1}{2+2s^2-2s^4} \\ \frac{(-1+s)s}{4(-1-s^2+s^4)} & \frac{1}{2+2s^2-2s^4} & \frac{(-1+s)s}{4(-1-s^2+s^4)} \end{bmatrix}.$$

Kako je uslov $A(s)A(s)^\dagger = A(s)^\dagger A(s)$ zadovoljen a indeks za $A(s)$ je 1, zaključujemo da je $A(s)^\dagger = A(s)^D = A(s)^\#$. Zaista, matrica $A(s)^D$ je dobijena u slučaju $R(s) = A(s)^T$, $T(s) = A(s)$. Drazin inverz za $A(s)$ je poznat iz [20].

U slučaju $R(s) = A(s)$, $T(s) = I_3$ Dobijamo matricu

$$X_1(s) = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}.$$

Jednakostr $X_1(s) = A(s)A(s)^D$ može biti verifikovana posle primene standardne MATHEMATICA funkcije $Simplify[\,]$. Ovo potvrđuje deo (iv) u Teoremi 2.21.

Primer 2.23 Neka je data sledeća test matrica iz [86]

$$F4(s) = \begin{bmatrix} s+4 & s+3 & s+2 & s+1 \\ s+3 & s+3 & s+2 & s+1 \\ s+2 & s+2 & s+1 & s \\ s+1 & s+1 & s & s-1 \end{bmatrix}.$$

U slučaju $T(s) = R(s) = F4(s)$ dobija se poznati Moore-Penrose inverz za $F4(s)$ iz [86]:

$$F4(s)^{\dagger} = \begin{bmatrix} 1 & -\frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \\ -\frac{5}{6} & \frac{7}{9} - \frac{s}{4} & \frac{4}{9} & \frac{1}{9} + \frac{s}{4} \\ -\frac{1}{3} & \frac{4}{9} & \frac{1}{9} & -\frac{2}{9} \\ \frac{1}{6} & \frac{1}{9} + \frac{s}{4} & -\frac{2}{9} & -\frac{5}{9} - \frac{s}{4} \end{bmatrix}.$$

Koristeći

$$G(s) = \begin{bmatrix} 4 & s+3 & s+2 & s+1 \\ 0 & 0 & 0 & 0 \\ s+2 & s+2 & s+1 & s \\ s+1 & s+1 & s & s-1 \end{bmatrix}, \; H(s) = \begin{bmatrix} 4 & s+3 & s+2 & s+1 \\ 0 & 0 & 0 & 0 \\ s+2 & s+2 & s+1 & s \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

i $R(s) = Transpose[Simplify[G(s)H(s)]]$, $T(s) = F_4(s)$, dobija se inverz za $F4(s)$:

$$\begin{bmatrix} \frac{2(-4+4s-5s^2+5s^3+3s^5)}{-58+38s-59s^2+22s^3-9s^4+6s^5} & \frac{s(8-3s+7s^2-4s^3-2s^4)}{-58+38s-59s^2+22s^3-9s^4+6s^5} & \frac{4-2s+6s^2+s^3+2s^4+2s^5}{58-38s+59s^2-22s^3+9s^4-6s^5} & \frac{8+4s+9s^2+9s^3+2s^5}{58-38s+59s^2-22s^3+9s^4-6s^5} \\ 0 & 0 & 0 & 0 \\ \frac{-4+2s+4s^2+s^3-6s^4-3s^5}{-58+38s-59s^2+22s^3-9s^4+6s^5} & \frac{s(4-14s-2s^2+5s^3+s^4)}{-58+38s-59s^2+22s^3-9s^4+6s^5} & \frac{-2+s^2-s^3+4s^4+s^5}{-58+38s-59s^2+22s^3-9s^4+6s^5} & \frac{-4-4s+16s^2+3s^4+s^5}{-58+38s-59s^2+22s^3-9s^4+6s^5} \\ \frac{-12+10s+9s^2+5s^3+3s^4+3s^5}{-58+38s-59s^2+22s^3-9s^4+6s^5} & \frac{-29+7s+2s^2-3s^3+4s^4+s^5}{-58+38s-59s^2+22s^3-9s^4+6s^5} & \frac{6-2s+12s^2-s^3+3s^4+s^5}{-58+38s-59s^2+22s^3-9s^4+6s^5} & \frac{41-11s+22s^2+s^3+2s^4+s^5}{-58+38s-59s^2+22s^3-9s^4+6s^5} \end{bmatrix}.$$

Primer 2.24 Izaberimo sledeću matricu $A(s)$:

$$A(s) = T(s) = \begin{bmatrix} s+s^2 & 1 & s+s^2 \\ 1 & -s+s^2 & 1 \end{bmatrix}$$

i matricu

$$R(s) = \begin{bmatrix} s^3+s & s & s^2 \\ s-1 & s^2-s & 3 \end{bmatrix}$$

koja obezbeđuje invertibilnost matrice $A(s)R(s)^*$. Saglasno delu (iii) Teoreme 2.21 dobija se sledeći desni inverz za $A(s)$:

$$\begin{bmatrix} \frac{3+5s^2-3s^3+2s^4-2s^5+s^6}{2+2s+2s^2+2s^3-3s^4-2s^5-s^6+s^8} & \frac{1+s+5s^2+2s^3+3s^4}{-2-2s-2s^2-2s^3+3s^4+2s^5+s^6-s^8} \\ \frac{1}{1+s^2-s^4} & \frac{s(1+s)}{-1-s^2+s^4} \\ \frac{-3+2s-5s^2+s^3-2s^4+s^5}{2+2s+2s^2+2s^3-3s^4-2s^5-s^6+s^8} & \frac{3+3s+5s^2+2s^3+2s^4}{2+2s+2s^2+2s^3-3s^4-2s^5-s^6+s^8} \end{bmatrix}.$$

2.2 INVERZI POLINOMIJALNIH MATRICA

Neka je matrica $A = A[s]$ dimenzije $m \times n$ data u polinomijalnoj formi sa nepoznatim parametrom s:

$$A[s] = A_1 + A_2 s + \cdots + A_q s^{q-1} + A_{q+1} s^q = \sum_{i=0}^{q} A_{i+1} s^i \tag{2.2.1}$$

gde su A_i, $i = 1, \ldots, q+1$ konstantne matrice dimenzije $m \times n$.

2.2.1 Metoda pregradjivanja za polinomijalne matrice

Teorema 2.22 *Neka je $A[s]$ proizvoljna matrica dimenzije $m \times n$ data u polinomijalnoj formi* (2.2.1)*. Označimo i-tu kolonu iz $A[s]$ sa*

$$a_i(s) = \sum_{j=0}^{q} A_{i,j+1} s^j, \ \ 1 \leq i \leq n,$$

gde su $A_{i,j+1}$, $0 \leq j \leq q$ konstantni vektori dimenzije $m \times 1$. Takođe, označimo prvih i kolona iz $A[s]$ sa

$$\widehat{A}_i(s) = \sum_{j=0}^{q} \widehat{A}_{i,j+1} s^j, \ \ 1 \leq i \leq n,$$

gde su $\widehat{A}_{i,j+1}$, $0 \leq j \leq q$ konstantne matrice dimenzije $m \times i$. Za svako $i = 1, \ldots, n$ razmotrimo matrice $\widehat{A}_i(s)^\dagger$ koje su definisane sa

$$\widehat{A}_i(s)^\dagger = \frac{\sum\limits_{j=0}^{q_i} X_{i,j+1} s^j}{\sum\limits_{j=0}^{p_i} Y_{i,j+1} s^j}, \tag{2.2.2}$$

gde su

$$q_i = \begin{cases} 8q + 5q_{i-1}, & c_i(s) \neq 0, \\ 4q + 5q_{i-1}, & c_i(s) = 0, \end{cases}, \ p_i = q_i + q, \ i \geq 2, \ q_1 = q, \ p_1 = 2q, \tag{2.2.3}$$

kao i matrice $X_{i,j+1}$ i $Y_{i,j+1}$ definisane sa

$$\begin{aligned} X_{1,j+1} &= A^*_{1,j+1}, \quad 0 \leq j \leq q, \\ Y_{1,j+1} &= \sum_{k=0}^{j} A^*_{1,j-k+1} A_{1,k+1}, \quad 0 \leq j \leq 2q. \end{aligned} \tag{2.2.4}$$

$$X_{i,j+1} = \begin{bmatrix} \sum\limits_{k=0}^{j} \Theta_{i,j-k+1} W^*_{i,k+1} \\ \\ \sum\limits_{k=0}^{j} V^*_{i,j-k+1} \Phi_{i,k+1} \end{bmatrix}, \quad 0 \le j \le q_i, \ i \ge 2 \tag{2.2.5}$$

$$Y_{i,j+1} = \begin{cases} \sum\limits_{k=0}^{j} \Phi_{i,j-k+1} W^*_{i,k+1}, & a_1(s) \neq 0, \\ 1, & a_1(s) = 0, \end{cases} \quad 0 \le j \le p_i, \ i \ge 2$$

gde su u (2.2.5)

$$\Theta_{i,j+1} = \sum_{k=0}^{j} (X_{i-1,j-k+1} W^*_{i,k+1} - d_{i,j-k+1} V^*_{i,k+1}), \quad 0 \le j \le l_1,$$

$$\Phi_{i,j+1} = \sum_{k=0}^{j} Y_{i-1,j-k+1} W^*_{i,k+1}, \qquad 0 \le j \le l_2, \tag{2.2.6}$$

$$l_1 = \begin{cases} 4q + 3q_{i-1}, & c_i(s) \neq 0, \\ 2q + 3q_{i-1}, & c_i(s) = 0, \end{cases}, \quad l_2 = l_1 + q,$$

kao i u (2.2.6)

$$d_{i,j+1} = \sum_{k=0}^{j} X_{i-1,j-k+1} A_{i,k+1}, \quad 0 \le j \le q + q_{i-1}, \tag{2.2.7}$$

matrice $V_{i,j+1}$, $W_{i,j+1}$ *definisane sa sledećim pravilima:*
Izračunati

$$c_{i,j+1} = \sum_{k=0}^{j} (A_{i,j-k+1} Y_{i-1,k+1} - \widehat{A}_{i-1,j-k+1} d_{i,k+1}), \quad 0 \le j \le 2q + q_{i-1}. \tag{2.2.8}$$

U slučaju da su $c_{i,j+1} \neq 0$ *za neko* j, $V_{i,j+1}$ *i* $W_{i,j+1}$ *jednaki sa*

$$V_{i,j+1} = \sum_{k=0}^{j} c_{i-1,j-k+1} Y^*_{i,k+1}, \quad 0 \le j \le 3q + 2q_{i-1} \tag{2.2.9}$$

$$W_{i,j+1} = \sum_{k=0}^{j} c^*_{i,j-k+1} c_{i,k+1}, \quad 0 \le j \le 4q + 2q_{i-1};$$

u suprotnom, ako su $c_{i,j+1} = 0$ *za svako* j, $V_{i,j+1}$ *i* $W_{i,j+1}$ *definisane sa*

$$V_{i,j+1} = \sum_{k=0}^{j} X^*_{i-1,j-k+1} d_{i,k+1}, \ 0 \le j \le q + 2q_{i-1}, \tag{2.2.10}$$

$$W_{i,j+1} = \sum_{k=0}^{j} \left(Y^*_{i-1,j-k+1} Y_{i-1,k+1} - d^*_{i,j-k+1} d_{i,k+1} \right), \ 0 \le j \le 2q + 2q_{i-1}.$$

Onda je Moore-Penrose inverz $A(s)^\dagger$ za $A(s)$ jednak $\widehat{A}_n(s)^\dagger$.

Dokaz. Jasno je da se i-ta kolona $a_i(s)$ u $A(s)$ može predstaviti u polinomijalnoj formi $a_i(s)=\sum_{j=0}^{q} A_{i,j+1}s^j$, $1\leq i\leq n$, gde su $A_{i,j+1}$, $0 \leq j \leq q$ konstantni vektori dimenzije $m \times 1$. Ako je $a_1(s) \neq 0$, saglasno *Koraku 1* u algoritma osnovnog metoda pregrađivanja imamo

$$\widehat{A}_1(s)^\dagger = \frac{\sum_{j=0}^{q} A^*_{1,j+1}s^j}{\sum_{j=0}^{q} A^*_{1,j+1}s^j \sum_{j=0}^{q} A_{1,j+1}s^j} = \frac{\sum_{j=0}^{q} A^*_{1,j+1}s^j}{\sum_{j=0}^{2q}(\sum_{k=0}^{j} A^*_{1,j-k+1}A_{1,k+1})s^j}.$$

Tim pre, možemo uočiti da je $\widehat{A}_1(s)^\dagger$ specijalan slučaj za $i = 1$ u (2.2.2), gde su matrice $X_{1,j+1}$ i $Y_{1,j+1}$ definisane u (2.2.4).

Za svako $i = 2, \ldots, n$ razumljivo je izračunati matrice $\widehat{A}_i(s)^\dagger$ u formi (2.2.2), za odgovarajuće matrice $X_{i,j+1}$, $Y_{i,j+1}$ i odgovarajuće granice q_i i p_i.

Direktnim izračunavanjem u *Koraku 2.1* dobijamo

$$\begin{aligned} d_i(s) &= \widehat{A}_{i-1}(s)^\dagger a_i(s) = \frac{\sum_{j=0}^{q_{i-1}} X_{i-1,j+1}s^j}{\sum_{j=0}^{p_{i-1}} Y_{i-1,j+1}s^j} \cdot \sum_{k=0}^{q} A_{i,k+1}s^k \\ &= \frac{\sum_{j=0}^{q_{i-1}+q}(\sum_{k=0}^{j} X_{i-1,j-k+1}A_{i,k+1})s^j}{\sum_{j=0}^{p_{i-1}} Y_{i-1,j+1}s^j}. \end{aligned}$$

Onda se $d_i(s)$ može predstaviti u formi

$$d_i(s) = \frac{\sum_{j=0}^{q_{i-1}+q} d_{i,j+1}s^j}{\sum_{j=0}^{p_{i-1}} Y_{i-1,j+1}s^j}, \tag{2.2.11}$$

gde su matrice $d_{i,j+1}$ definisane sa (2.2.7).

Razmotrimo *Korak 2.2*. Kako se prvih $i-1$ kolona u A može predstaviti u polinomijalnoj formi

$$\widehat{A}_{i-1}(s) = \sum_{j=0}^{q} \widehat{A}_{i-1,j+1}s^j$$

za odgovarajuće $m \times (i-1)$ konstantne matrice $\widehat{A}_{i-1}(s)$, saglasno (2.2.11) dobija se

$$
\begin{aligned}
c_i(s) &= a_i(s) - \widehat{A}_{i-1}(s) d_i(s) = \sum_{j=0}^{q} A_{i,j+1} s^j - \sum_{j=0}^{q} \widehat{A}_{i-1,j+1} s^j \cdot \frac{\sum\limits_{j=0}^{q_{i-1}+q} d_{i,j+1} s^j}{\sum\limits_{j=0}^{p_{i-1}} Y_{i-1,j+1} s^j} \\
&= \frac{\sum\limits_{j=0}^{q+p_{i-1}} \left(\sum\limits_{k=0}^{j} A_{i,j-k+1} Y_{i-1,k+1} \right) s^j - \sum\limits_{j=0}^{2q+q_{i-1}} (\sum\limits_{k=0}^{j} \widehat{A}_{i-1,j-k+1} d_{i,k+1}) s^j}{\sum\limits_{j=0}^{q_{i-1}+q} Y_{i-1,j+1} s^j}
\end{aligned}
$$

Izjednačavanjem gornjih granica u poslednjim jednakostima dobijamo da je $p_{i-1} = q_{i-1} + q$, i

$$
c_i(s) = \frac{\sum\limits_{j=0}^{2q+q_{i-1}} (\sum\limits_{k=0}^{j} (A_{i,j-k+1} Y_{i-1,k+1} - \widehat{A}_{i-1,j-k+1} d_{i,k+1})) s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1} s^j}.
$$

Tim pre, $c_i(s)$ se može predstaviti u formi

$$
c_i(s) = \frac{\sum\limits_{j=0}^{2q+q_{i-1}} c_{i,j+1} s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1} s^j},
$$

gde su $c_{i,j+1}$ matrice iz (2.2.8), za svako $0 \le j \le 2q + q_{i-1}$.

Posmatrajmo sad *Korak 2.3.* Ako je $c_{i,j+1} \neq 0$ za neko j, onda je $c_i(s) \neq 0$ i $b_i(s)$, je jednako

$$
\begin{aligned}
b_i(s) &= \frac{c_i(s)}{c_i(s)^* c_i(s)} = \frac{\sum\limits_{j=0}^{2q+q_{i-1}} c_{i,j+1} s^j \cdot \sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1}^* s^j}{\sum\limits_{j=0}^{2q+q_{i-1}} c_{i,j+1}^* s^j \cdot \sum\limits_{j=0}^{2q+q_{i-1}} c_{i,j+1} s^j} \\
&= \frac{\sum\limits_{j=0}^{3q+2q_{i-1}} (\sum\limits_{k=0}^{j} c_{i-1,j-k+1} Y_{i,k+1}^*) s^j}{\sum\limits_{j=0}^{4q+2q_{i-1}} (\sum\limits_{k=0}^{j} c_{i,j+1}^* c_{i,k+1}) s^j}. \qquad (2.2.12)
\end{aligned}
$$

Jednakost (2.2.12) može se napisati u formi

$$b_i(s) = \frac{\sum\limits_{j=0}^{3q+2q_{i-1}} V_{i,j+1}s^j}{\sum\limits_{j=0}^{4q+2q_{i-1}} W_{i,j+1}s^j}$$

gde su $V_{i,j+1}$ i $W_{i,j+1}$ matrice definisane u (2.2.9).

Ako je $c_{i,j+1} = 0$ za svako j, imamo da je $c_i(s) = 0$, i koristeći predhodni rezultat može se verifikovati sledeće

$$\begin{aligned}
b_i(s) &= \frac{(A_{i-1}(s)^\dagger)^* d_i(s)}{1 + d_i(s)^* d_i(s)} = \frac{\dfrac{\sum\limits_{j=0}^{q_{i-1}} X^*_{i-1,j+1}s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y^*_{i-1,j+1}s^j} \cdot \dfrac{\sum\limits_{j=0}^{q+q_{i-1}} d_{i,j+1}s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1}s^j}}{1 + \dfrac{\sum\limits_{j=0}^{q+q_{i-1}} d^*_{i,j+1}s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y^*_{i-1,j+1}s^j} \cdot \dfrac{\sum\limits_{j=0}^{q+q_{i-1}} d_{i,j+1}s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1}s^j}} \\
&= \frac{\sum\limits_{j=0}^{q+2q_{i-1}} (\sum\limits_{k=0}^{j} X^*_{i-1,j-k+1} d_{i,k+1}) s^j}{\sum\limits_{j=0}^{2q+2q_{i-1}} (\sum\limits_{k=0}^{j} Y^*_{i-1,j-k+1} Y_{i-1,k+1} + d^*_{i,j-k+1} d_{i,k+1}) s^j} \\
&= \frac{\sum\limits_{j=0}^{q+2q_{i-1}} V_{i,j+1}s^j}{\sum\limits_{j=0}^{2q+2q_{i-1}} W_{i,j+1}s^j},
\end{aligned}$$

gde su $V_{i,j+1}$ i $W_{i,j+1}$ matrice definisane u (2.2.10).

Konačno, primenom *Koraka 2* i prethodnih rezultata u slučaju $c_i(s) \neq 0$

dobija se

$$
\begin{aligned}
\widehat{A}_i(s)^\dagger &= \begin{bmatrix} \widehat{A}_{i-1}(s)^\dagger - 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+1}s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1}s^j} - \dfrac{\sum\limits_{j=0}^{q+q_{i-1}} d_{i,j+1}s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1}s^j} \cdot \dfrac{\sum\limits_{j=0}^{3q+2q_{i-1}} V^*_{i,j+1}s^j}{\sum\limits_{j=0}^{4q+2q_{i-1}} W^*_{i,j+1}s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{3q+2q_{i-1}} V^*_{i,j+1}s^j}{\sum\limits_{j=0}^{4q+2q_{i-1}} W^*_{i,j+1}s^j} \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{\sum\limits_{j=0}^{4q+3q_{i-1}} (\sum\limits_{k=0}^{j} (X_{i-1,j-k+1}W^*_{i,k+1} - d_{i,j-k+1}V^*_{i,k+1}))s^j}{\sum\limits_{j=0}^{5q+3q_{i-1}} (\sum\limits_{k=0}^{j} Y_{i-1,j-k+1}W^*_{i,k+1})s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{3q+2q_{i-1}} V^*_{i,j+1}s^j}{\sum\limits_{j=0}^{4q+2q_{i-1}} W^*_{i,j+1}s^j} \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{\sum\limits_{j=0}^{4q+3q_{i-1}} \Theta_{i,j+1}s^j}{\sum\limits_{j=0}^{5q+3q_{i-1}} \Phi_{i,j+1}s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{3q+2q_{i-1}} V^*_{i,j+1}s^j}{\sum\limits_{j=0}^{4q+2q_{i-1}} W^*_{i,j+1}s^j} \end{bmatrix} = \begin{bmatrix} \dfrac{\sum\limits_{j=0}^{4q+3q_{i-1}} \Theta_{i,j+1} \sum\limits_{j=0}^{4q+2q_{i-1}} W^*_{i,j+1}s^j}{\sum\limits_{j=0}^{5q+3q_{i-1}} \Phi_{i,j+1} \sum\limits_{j=0}^{4q+2q_{i-1}} W^*_{i,j+1}s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{3q+2q_{i-1}} V^*_{i,j+1} \sum\limits_{j=0}^{5q+3q_{i-1}} \Phi_{i,j+1}s^j}{\sum\limits_{j=0}^{4q+2q_{i-1}} W^*_{i,j+1} \sum\limits_{j=0}^{5q+3q_{i-1}} \Phi_{i,j+1}s^j} \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{\sum\limits_{j=0}^{8q+5q_{i-1}} (\sum\limits_{k=0}^{j} \Theta_{i,j-k+1}W^*_{i,k+1})s^j}{\sum\limits_{j=0}^{9q+5q_{i-1}} (\sum\limits_{k=0}^{j} \Phi_{i,j-k+1}W^*_{i,k+1})s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{8q+5q_{i-1}} (\sum\limits_{k=0}^{j} V^*_{i,j-k+1}\Phi_{i,k+1})s^j}{\sum\limits_{j=0}^{9q+5q_{i-1}} (\sum\limits_{k=0}^{j} \Phi_{i,j-k+1}W^*_{i,k+1})s^j} \end{bmatrix} = \frac{\sum\limits_{j=0}^{q_i} X_{i,j+1}s^j}{\sum\limits_{j=0}^{p_i} Y_{i,j+1}s^j},
\end{aligned}
$$

gde $X_{i,j+1}$ i $Y_{i,j+1}$ zadovoljavaju (2.2.5) a gornje granice q_i, p_i zadovoljavaju (2.2.3).

Primenom *Koraka 2* u slučaju $c_i(s) = 0$ dobija se

$$
\begin{aligned}
\widehat{A}_i(s)^\dagger &= \begin{bmatrix} \dfrac{\sum\limits_{j=0}^{q_{i-1}} X_{i-1,j+1}s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1}s^j} - \dfrac{\sum\limits_{j=0}^{q+q_{i-1}} d_{i,j+1}s^j}{\sum\limits_{j=0}^{q+q_{i-1}} Y_{i-1,j+1}s^j} \cdot \dfrac{\sum\limits_{j=0}^{q+2q_{i-1}} V^*_{i,j+1}s^j}{\sum\limits_{j=0}^{2q+2q_{i-1}} W^*_{i,j+1}s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{q+2q_{i-1}} V^*_{i,j+1}s^j}{\sum\limits_{j=0}^{2q+2q_{i-1}} W^*_{i,j+1}s^j} \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{\sum\limits_{j=0}^{2q+3q_{i-1}} \Theta_{i,j+1}s^j}{\sum\limits_{j=0}^{3q+3q_{i-1}} \Phi_{i,j+1}s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{q+2q_{i-1}} V^*_{i,j+1}s^j}{\sum\limits_{j=0}^{2q+2q_{i-1}} W^*_{i,j+1}s^j} \end{bmatrix} = \begin{bmatrix} \dfrac{\sum\limits_{j=0}^{2q+3q_{i-1}} \Theta_{i,j+1} \sum\limits_{j=0}^{2q+2q_{i-1}} W^*_{i,j+1}s^j}{\sum\limits_{j=0}^{3q+3q_{i-1}} \Phi_{i,j+1} \sum\limits_{j=0}^{2q+2q_{i-1}} W^*_{i,j+1}s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{q+2q_{i-1}} V^*_{i,j+1} \sum\limits_{j=0}^{3q+3q_{i-1}} \Phi_{i,j+1}s^j}{\sum\limits_{j=0}^{2q+2q_{i-1}} W^*_{i,j+1} \sum\limits_{j=0}^{3q+3q_{i-1}} \Phi_{i,j+1}s^j} \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{\sum\limits_{j=0}^{4q+5q_{i-1}} (\sum\limits_{k=0}^{j} \Theta_{i,j-k+1}W^*_{i,k+1})s^j}{\sum\limits_{j=0}^{5q+5q_{i-1}} (\sum\limits_{k=0}^{j} \Phi_{i,j-k+1}W^*_{i,k+1})s^j} \\ \\ \dfrac{\sum\limits_{j=0}^{4q+5q_{i-1}} (\sum\limits_{k=0}^{j} V^*_{i,j-k+1}\Phi_{i,k+1})s^j}{\sum\limits_{j=0}^{5q+5q_{i-1}} (\sum\limits_{k=0}^{j} \Phi_{i,j-k+1}W^*_{i,k+1})s^j} \end{bmatrix} = \dfrac{\sum\limits_{j=0}^{q_i} X_{i,j+1}s^j}{\sum\limits_{j=0}^{p_i} Y_{i,j+1}s^j},
\end{aligned}
$$

gde ponovo $X_{i,j+1}$ i $Y_{i,j+1}$ zadovoljavaju (2.2.5) a gornje granice q_i, p_i zadovoljavaju (2.2.3).

Na kraju, Moore-Penroseov inverz date matrice je $A(s)^\dagger = \widehat{A}_n(s)^\dagger$, čime je dokaz završen. $\square$

U skladu sa prethodnom teoremom može se predložiti sledeći algoritam.

Algoritam 2.6 *Ulazni parametar je polinomijalna matrica* $A[s]$ *u formi* (2.2.1), *čija je* i*-ta kolona predstavljena u formi* $a_i(s) = \sum\limits_{j=0}^{q} A_{i,j+1}s^j$, $1 \leq i \leq n$, *Korak 1. Inicijalne vrednosti:*

Izračunati $X_{1,j+1}$, $0 \le j \le q$ *i* $Y_{1,j+1}$, $0 \le j \le 2q$ *kao u* (2.2.4).

Korak 2. Rekurzivni korak:

Za $2 \le i \le n$ *izvršiti Korak 2.1, Korak 2.2 i Korak 2.3.*

Korak 2.1. *Izračunati* $d_{i,j+1}$, $0 \le j \le q + q_{i-1}$ *u skladu sa* (2.2.7).

Korak 2.2. *Izračunati* $c_{i,j+1}$, $0 \le j \le 2q + q_{i-1}$ *kao u* (2.2.8).

Korak 2.3. *Ako je* $c_{i,j+1} \neq 0$ *za neko* j, *izračunati* $V_{i,j+1}$ *i* $W_{i,j+1}$ *kao u* (2.2.9). *U slučaju da je* $c_{i,j+1} = 0$ *za svako* j, *izračunati* $V_{i,j+1}$ *i* $W_{i,j+1}$ *kao u* (2.2.10).
Onda izračunati

$$X_{i,j+1},\ 0 \le j \le q_i, \quad Y_{i,j+1},\ 0 \le j \le q_i + q$$

kao što je definisano u (2.2.5).

Korak 3. Kriterijum zaustavljanja je $i = n$. *U tom slučaju je* $A(s)^\dagger = \widehat{A}_n(s)^\dagger$.

U tekstu koji sledi opisana su glavni koraci u implementaciji algoritma. Matrica $A[s]$ definisana u (2.2.1) može se reprezentovati kao lista matrica $\{A_1, \ldots, A_{q+1}\}$. Slično, i-ta kolona $A_i(s)$ iz $A[s]$ je polinomijalna matrica sa parametrom s:

$$A_i(s) = \sum_{j=0}^{q} A_{i,j+1} s^j$$

pa se tim pre može reprezentovati kao tro-dimenzionalna lista $\{A_{i,1}, \ldots, A_{i,q+1}\}$, $1 \le i \le n$. Ovakvu reprezentaciju trebamo imati u vidu kad opisujemo implementaciju algoritma. Takođe, ograničićemo implementaciju na skup polinomijalnih matrica sa realnim koeficijentima u cilju pojednostavljenja koristeći standardnu MATHEMATICA funkciju $Simplify[\,]$. Nekoliko pomoćnih procedura je opisano na početku.

A. Funkcija *FrmPoly*[*M*] transformiše matricu $M(s) = \sum_{j=0}^{q} M_{j+1} s^j$ u internu reprezentaciju $\{M_1, \ldots, M_{q+1}\}$.

```
FrmPoly[M_List]:=
   Module[{L={},i,M1=M,v,s},
     v=Variables[M];
```

```
    If[v=!={},s=v[[1]];
    For[i=1,i<=Max[Exponent[M,s]],i++,
        AppendTo[L,Coefficient[M,s^i]];
        M1=M1-Coefficient[M,s^i]*s^i;
    ];
    M1={M1};
    For[i=1,i<=Length[L],i++, AppendTo[M1,L[[i]]]];
    ];
    Return[Simplify[M1]];
  ];
```

B. Funkcija $Col[L, j]$, listu j-tih kolona iz matrice, predstavljenu kao tro-dimenzionalna lista L transformiše takođe u tro-dimenzionalnu listu.

```
Col[L_List,j_]:=
  Block[{L1=L2={},i},
    For[i=1,i<=Length[L],i++,
        L1=Append[L1,Transpose[L[[i]]]];
        AppendTo[L2,L1[[i,j]]]
    ];
    Return[L2]
  ];
```

C. Rezultat funkcije $TakeFPoly[L, j]$ je tro-dimenzionalna lista koja sadrži prvih j kolona iz tro-dimenzionalne liste L.

```
TakeFPoly[L_List,j_]:=
  Module[{L1={},i},
    For[i=1,i<=Length[L],i++,
        L1=Append[L1,Take[Transpose[L[[i]]],j]]];
    Return[L1]
  ];
```

D. Funkcija $DopZero[L, i]$ dopunjuje tro-dimenzionalnu listu L nula matricama od pozicije $Length[L]+1$ do i.

```
DopZero[L_List,i_]:=
  Module[{L1=L,j,nul},
    nul=L[[1]]*0;
    For[j=1,j<=i-Length[L],j++,
        AppendTo[L1,nul]
    ];
    Return[L1]
  ];
```

E. Funkcija $LastZeroP[L]$ briše poslednje nula elemente iz tro-dimenzionalne liste L.

```
LastZeroP[L_List]:=
   Module[{L1=L,Us=True,nul,dl},
     If[L1=!={},
        While[Us && L1=!={},dl=Dimensions[L1][[1]];
          If[L1[[dl]]==L1[[dl]]*0,  L1=Drop[L1,-1],  Us=False]
     ]  ];
     Return[L1]
   ];
```

Implementacija *Koraka 2.1.*

```
DDP[L_List,i_]:=DDP[L,i]=
   Module[{Y={},bound=0,NN,L2={},L1=L,j,nul={}},
     L2=XXP[L,i-1];
     bound=Length[L]+Length[L2];
     nul={Table[0,{j,1,bound}]};
     L1=DopZero[L1,bound];
     NN=Col[L1,i];
     L2=DopZero[L2,bound];
     For[j=0,j<bound-1,j++,
        If[(j+1)>Length[Y],Y=Join[Y,nul] ];
        Y[[j+1]]=Sum[L2[[j-k+1]].NN[[k+1]],{k,0,j}];
     ];
     Y=LastZeroP[Y];
     Return[Y]
   ];
```

Implementacija *Koraka 2.2.*

```
CCP[L_List,i_]:=CCP[L,i]=
   Module[{Y=L4={},bound=0,NN,L1=L,L2=L3={},j,nul={}},
     L2=YYP[L,i-1];
     bound=2Length[L]+Length[L2];
     nul=Table[0,{j,1,bound}];
     L1=DopZero[L1,bound];
     NN=Col[L1,i];
     L2=DopZero[L2,bound];
     L4=DDP[L,i];  If[L4=={}, L4={0}];      L4=DopZero[L4,bound];
     L3=TakeFPoly[L,i-1]; L3=DopZero[L3,bound];
     For[j=0,j<bound-1,j++,
      If[(j+1)>Length[Y],Y=Join[Y,nul]];
      If[Length[L4[[1]]]==0,
```

```
        Y[[j+1]]=Sum[NN[[j-k+1]]L2[[k+1]]
                -(L3[[j-k+1]]L4[[k+1]])[[1]],{k,0,j}],
        Y[[j+1]]=Sum[NN[[j-k+1]]L2[[k+1]]
                -Transpose[L3[[j-k+1]]].L4[[k+1]],{k,0,j}]
      ]];
      Return[LastZeroP[Y]];
    ];
```

Implementacija *Koraka 2.3.*

```
VVP[L_List,i_]:=VVP[L,i]=
   Module[{Y={},bound=0,L1=L2=L3=L4={},j,nul={}},
     L1=YYP[L,i-1];L2=CCP[L,i];
     If[L2=!={},
        bound=3Length[L]+2Length[L1];
        nul=Table[0,{j,1,bound}];
        L1=DopZero[L1,bound]; L2=DopZero[L2,bound];
        For[j=0,j< bound-1,j++,
           If[(j+1)>Length[Y], Y=Join[Y,nul] ];
           Y[[j+1]]=Sum[L1[[j-k+1]]L2[[k+1]],{k,0,j}];
        ],
        L3=XXP[L,i-1];L4=DDP[L,i];
        bound=Length[L]+2Length[L3];
        nul=Table[0,{j,1,bound}];
        L3=DopZero[L3,bound];
        L4=DopZero[L4,bound];
        For[j=0,j<bound-1,j++,
         If[(j+1)>Length[Y], Y=Join[Y,nul] ];
         If[Length[L4[[1]]]==0,
          Y[[j+1]]=Sum[L3[[j-k+1]]L4[[k+1]],{k,0,j}],
          Y[[j+1]]=Sum[Transpose[L3[[j-k+1]]].L4[[k+1]],{k,0,j}]
     ]  ] ];
     Return[LastZeroP[Y]];
];
```

```
WWP[L_List,i_]:=WWP[L,i]=
   Module[{Y={},bound=0,L1=L2=L3={},j,nul={}},
     L1=CCP[L,i];   L2=YYP[L,i-1];
     If[L1=!={},
       bound=4Length[L]+2Length[L2];
       nul=Table[0,{j,1,bound}];
       L1=DopZero[L1,bound];
       For[j=0,j<bound-1,j++,
```

```
          If[(j+1)>Length[Y], Y=Join[Y,nul] ];
           Y[[j+1]]=Sum[L1[[j-k+1]].L1[[k+1]],{k,0,j}];
        ],
        L3=DDP[L,i];
        bound=2(Length[L]+Length[L2]);
        nul=Table[0,{j,1,bound}];
        L2=DopZero[L2,bound];   L3=DopZero[L3,bound];
        For[j=0,j<bound-1,j++,
          If[(j+1)>Length[Y], Y=Join[Y,nul] ];
          If[Length[L3[[1]]]==0,
            Y[[j+1]]=Sum[L2[[j-k+1]]L2[[k+1]]
                    +L3[[j-k+1]]L3[[k+1]],{k,0,j}],
            Y[[j+1]]=Sum[L2[[j-k+1]]L2[[k+1]]+({L3[[j-k+1]]}
                      .Transpose[{L3[[k+1]]}])[[1]],{k,0,j}][[1]]
      ] ] ];
      Y=LastZeroP[Y];
      Return[Y];
    ];
XXP[L_List,i_]:=XXP[L,i]=
   Module[{Y=Y1={},dq,L0=L1=L2=L3=L4=L5=L6=L7={},
           j,iz={},k1,bound=0,nul={}},
      If[i==1,  (* Then *)
        nul=L[[1,1]]*0;  dq=Length[L];
        For[j=1,j<=dq,j++,
           Y1={};
           For[k=1,k<=Length[L[[1]]],k++,
             AppendTo[Y1,L[[j,k,1]]]
           ];
           AppendTo[Y,Y1]
        ];
        iz=Y,    (* Else *)
        L0=XXP[L,i-1];L1=XXP[L,i-1]; L6=YYP[L,i-1];
        L2=WWP[L,i];L3=DDP[L,i];L4=VVP[L,i];
        bound=4Length[L]+3Length[L0];
        nul=Table[0,{j,1,bound}];
        L1=DopZero[L1,bound-1];  L2=DopZero[L2,bound-1];
        If[L3=={}, L3={0}];  L3=DopZero[L3,bound-1];
        L4=DopZero[L4,bound-1];  Y={};
        For[j=0,j<bound-1,j++,
          If[(j+1)>Length[Y], Y=Join[Y,nul]];
```

```
          If[Length[L3[[1]]]==0,
           Y[[j+1]]=Sum[L1[[j-k+1]]L2[[k+1]]
                   -L3[[j-k+1]]L4[[k+1]],{k,0,j}],
           Y[[j+1]]=Sum[L1[[j-k+1]]L2[[k+1]]
            -Transpose[{L3[[j-k+1]]}].{L4[[k+1]]},{k,0,j}]
        ] ];
        L5=Y;bound=5Length[L]+3Length[L0];nul=Table[0,{j,1,bound}];
        L2=DopZero[L2,bound-1];L6=DopZero[L6,bound-1];Y={};
        For[j=0,j<bound-1,j++,
          If[(j+1)>Length[Y], Y=Join[Y,nul]];
          Y[[j+1]]=Sum[L6[[j-k+1]]L2[[k+1]],{k,0,j}];
        ];
        L7=Y; bound=8Length[L]+5Length[L0]; nul=Table[0,{j,1,bound}];
        L5=DopZero[L5,bound-1]; L7=DopZero[L7,bound-1];
        L2=DopZero[L2,bound-1]; L4=DopZero[L4,bound-1];
        Y={}; Y1={}; iz={};
        For[j=0,j<bound-1,j++,
           If[(j+1)>Length[Y], Y=Join[Y,nul];
           Y1=Join[Y1,nul];  iz=Join[iz,nul]];
           Y[[j+1]]=Sum[L2[[j-k+1]]L5[[k+1]],{k,0,j}];
           Y1[[j+1]]=Sum[L4[[j-k+1]]L7[[k+1]],{k,0,j}];
           If[Length[Dimensions[Y[[1]]]]==1,
             iz[[j+1]]=Join[{Y[[j+1]]},{Y1[[j+1]]}],
             iz[[j+1]]=Join[Y[[j+1]],{Y1[[j+1]]}]
           ]
        ];
        iz=LastZeroP[iz];
      ];
      Return[iz];
    ];
YYP[L_List,i_]:=YYP[L,i]=
   Module[{Y={},L1={},NN,bound=0,iz={},L0=L1=L2=L5=={},k1,j,nul={}},
     If[i==1, (* Then *)
        dq=2Length[L];  nul={0};
        L1=DopZero[L,dq];    NN=Col[L1,1];
        If[NN==NN*0,
          Y={1},
          For[j=0,j<dq-1,j++,
             If[(j+1)>Length[Y], Y=Join[Y,nul]];
             Y[[j+1]]=Sum[{NN[[j-k+1]]}.NN[[k+1]],{k,0,j}][[1]];
```

```
        ] ];
        iz=LastZeroP[Y],   (* Else *)
        L1=YYP[L,i-1];  L2=WWP[L,i];  L0=XXP[L,i-1];
        bound=5Length[L]+3Length[L0];  nul=Table[0,{j,1,bound}];
        L1=DopZero[L1,bound];  L2=DopZero[L2,bound];
        For[j=0,j<bound-1,j++,
          If[(j+1)>Length[Y],Y=Join[Y,nul]];
          Y[[j+1]]=Sum[L1[[j-k+1]]L2[[k+1]],{k,0,j}];
        ];
        L5=Y;  bound=9Length[L]+5Length[L0];  nul=Table[0,{j,1,bound}];
        L5=DopZero[L5,bound-1];  L2=DopZero[L2,bound-1];  Y={};
        For[j=0,j<bound-1,j++,
          If[(j+1)>Length[Y], Y=Join[Y,nul]];
          Y[[j+1]]=Sum[L2[[j-k+1]]L5[[k+1]],{k,0,j}];
        ];
        iz=LastZeroP[Y];
      ];
      Return[iz];
    ];
```

Funkcija *SimplP*[ ] u sebi sadrži pomoćnu funkciju *PolLCM*[ ]. Funkcija *PolLCM*[*L*] izračunava najmanji zajednički sadržalac polinoma sadržanih u listi L, i sadrži standardne MATHEMATICA funkcije *PolynomialLCM* i *LCM*.

```
SimplP[M1_List,M2_List]:=
   Module[{p,q,r,vr={},M3=M4={},i},
     p=Sum[M1[[i+1]]*w^i,{i,0,Length[M1]-1}];
     q=Sum[M2[[i+1]]*w^i,{i,0,Length[M2]-1}];
     r=Simplify[p/q];
     M3=PolLCM[Denominator[r]];
     If[Variables[M3]=!={},  (* Then *)
        M4=Expand[Simplify[r*M3]];
        M3=Transpose[FrmPoly[{M3}]][[1]];
        M4=FrmPoly[M4],  (* Else *)
        M4=FrmPoly[r];  M3={1}
     ];
     Return[{M4,M3}]
   ];

PolLCM[L_List]:=
   Module[{m=m1=1,j},
     For[j=1,j<=Length[L],j++,
        If[Variables[L]=!={},
```

```
        m=PolynomialLCM[m,L[[j]]],
        m=LCM[m,L[[j]]]
  ]  ];
  If[Variables[m]=={},
     If[Length[m]=!=0,  (* Then *)
       For[j=1,j<=Length[m],j++,
           m1=LCM[m1,m[[j]] ]
     ] ],              (* Else *)
     If[Not[Head[m]=!=List], (* Then *)
        For[j=1,j<=Length[m],j++,
          m1=PolynomialLCM[m1,m[[j]] ]
        ],                  (* Else *)
        m1=m
  ]  ];
  Return[Expand[m1]]
];
```

Rezultat funkcije $PartPoly[L]$ je Moore-Penroseov inverz za L.

```
PartPoly[L_List]:=
   Module[{m,n,i,res={},L1=L2=L3={}},
     {m,n}=Dimensions[L];
     A=FrmPoly[L];
     For[i=1,i<=n-1,i++,
       L1=XXP[A,i];  L2=YYP[A,i];
       res=SimplP[L1,L2];
       XXP[A,i]=res[[1]];   YYP[A,i]=res[[2]];
       VVP[A,i+1];      WWP[A,i+1];
     ];
     L1=XXP[A,n];   L2=YYP[A,n];
     res=SimplP[L1,L2];
     L2=Sum[res[[2,j]](Variables[L][[1]])^(j-1),{j,1,Length[res[[2]]]}];
     L1=Sum[res[[1,j]](Variables[L][[1]])^(j-1),{j,1,Length[res[[1]]]}];
     Return[Simplify[L1/L2]//MatrixForm]
   ];
```

Primer 2.25 Posmatrajmo sledeću racionalnu matricu

$$X1 = \begin{bmatrix} 1/w^2 & w & (w+1)/w^3 \\ w & w^2-1 & w \\ w+1 & 1/w & w+1 \end{bmatrix}.$$

Primenom funkcije $Partitioning[\,]$ dobijamo rezultat:

In[1]:= Partitioning[X1]

`Out[1]=` $\begin{bmatrix} -w^3 & \frac{-1-w+w^5+w^6}{w\left(-2-w+w^2+w^3\right)} & \frac{-1-w+w^2+w^3-w^5}{-2-w+w^2+w^3} \\ 0 & \frac{1+w}{-2-w+w^2+w^3} & \frac{w}{2+w-w^2-w^3} \\ w^3 & -\frac{-1+w^4+w^5}{-2-w+w^2+w^3} & \frac{w-w^3+w^5}{-2-w+w^2+w^3} \end{bmatrix}$

Primer 2.26 Za matricu

$$X1 = \begin{bmatrix} 1+w & w & 1+w \\ w & -1+w & w \\ 1+w & w & 1+w \end{bmatrix}$$

primenom funkcije $Partitioning[\,]$ dobijamo sledeći Moore-Penrose inverz za A:

In[2]:= Partitioning[X1]

DD[X1,2]= $\{\{\frac{w(1+3w)}{2+4w+3w^2}\}\}$

CC[X1,2]= $\{\{\frac{w}{2+4w+3w^2}\},\{-\frac{2(1+w)}{2+4w+3w^2}\},\{\frac{w}{2+4w+3w^2}\}\}$

B[X1,2]= $\{\{\frac{w}{2}\},\{-1-w\},\{\frac{w}{2}\}\}$

A[X1,2]= $\{\{\frac{1-w}{2},w,\frac{1-w}{2}\},\{\frac{w}{2},-1-w,\frac{w}{2}\}\}$

DD[X1,3]={{1},{0}}

CC[X1,3]={{0},{0},{0}}

B[X1,3]= $\{\{\frac{1-w}{4}\},\{\frac{w}{2}\},\{\frac{1-w}{4}\}\}$

A[X1,3]= $\{\{\frac{1-w}{4},\frac{w}{2},\frac{1-w}{4}\},\{\frac{w}{2},-1-w,\frac{w}{2}\},\{\frac{1-w}{4},\frac{w}{2},\frac{1-w}{4}\}\}$

`Out[2]=` $\begin{pmatrix} \frac{1-w}{4} & \frac{w}{2} & \frac{1-w}{4} \\ \frac{w}{2} & -1-w & \frac{w}{2} \\ \frac{1-w}{4} & \frac{w}{2} & \frac{1-w}{4} \end{pmatrix}$

Funkcija $PartPoly[X1]$ daje identičan rezultat:

In[3]:=PartPoly[X1]

XXP[A,1]={{1,0,1},{1,1,1}}

YYP[A,1]={2,4,3}

DDP[A,2]={0,1,3}

CCP[A,2]={{0,-2,0},{1,-2,1}}

VVP[A,2]={{0,-4,0},{2,-12,2},{4,-14,4},{3,-6,3}}

WWP[A,2]={4,8,6}

XXP[A,2]={{{16,0,16},{0,-32,0}},{{80,32,80},{16,-224,16}},{{168,192,168}, {96,-720,96}},{{152,528,152},{264,-1360,264}},{{-20,832,-20},{416,-1624,416}}, {{-180,792,-180},{396,-1224,396}},{{-162,432,-162},{216,-540,216}}, {{-54,108,-54},{54,-108,54}}}

YYP[A,2]={32,192,528,832,792,432,108}

DDP[A,3]={{32,0},{192,0},{528,0},{832,0},{792,0},{432,0},{108,0}}

CCP[A,3]={}

VVP[A,3]={{512,0,512},{5632,1024,5632},{29184,12288,29184},{92672,70656,92672}, {196480,256000,196480},{280704,648960,280704},{245504,1210368,245504}, {57088,1701376,57088},{-177696,1815552,-177696},{-298080,1460160,-298080}, {-253152,864000,-253152},{-132192,357696,-132192},{-40824,93312,-40824}, {-5832,11664,-5832}}

WWP[A,3]={2048,24576,141312,512000,1297920,2420736,3402752,3631104,2920320, 1728000,715392,186624,23328}

XXP[A,3]={{{33554432,0,33554432},{0,-134217728,0},{33554432,0,33554432}} ,

{{973078528,67108864,973078528},{67108864,-4160749568,67108864},
{973078528,67108864,973078528}},{{13841203200,2013265920,13841203200},
{2013265920,-63417876480,2013265920},{13841203200,2013265920,13841203200}},
{{128429588480,29695672320,128429588480},{29695672320,-632501043200,29695672320},
{128429588480,29695672320,128429588480}},$\cdots$,{
{{-2595812935680,5779357102080,-2595812935680},
{5779357102080,-12734176665600,5779357102080},
{-2595812935680,5779357102080,-2595812935680}},
{{-279172334592,587731230720,-279172334592},
{587731230720,-1234235584512,587731230720},
{-279172334592,587731230720,-279172334592}},
{{-14693280768,29386561536,-14693280768},{29386561536,-58773123072,29386561536},
{-14693280768,29386561536,-14693280768}}}

YYP[A,3]={134217728,4026531840,59391344640,573109698560,4062267310080,
22516164722688,101458619924480,381515905105920,1219595599872000,
3359626347151360,8056217096159232,16944581899714560,
31436930673541120,51657089860239360,75387534830469120,
97868724073136128,113081302245703680,116228452185538560,
106099641023201280,85781945867304960,61176898573959168,38268243860520960,
20837934194688000,9777835364843520,3900410187448320,1298395518271488,
351376204677120,74358884597760,11558714204160,1175462461440,58773123072}

$$\texttt{Out[3]=}\begin{bmatrix} \frac{1-w}{4} & \frac{w}{2} & \frac{1-w}{4} \\ \frac{w}{2} & -1-w & \frac{w}{2} \\ \frac{1-w}{4} & \frac{w}{2} & \frac{1-w}{4} \end{bmatrix}$$

Mogu se posmatrati i nekoliko test matrica iz [86].

Primer 2.27 Test matrica [86]:

$$A=\begin{bmatrix}
s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s^2 & s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s^3 & s^2 & s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s^4 & s^3 & s^2 & s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s^5 & s^4 & s^3 & s^2 & s & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
s^6 & s^5 & s^4 & s^3 & s^2 & s & 1 & 0 & 0 & 0 & 0 & 0 \\
s^7 & s^6 & s^5 & s^4 & s^3 & s^2 & s & 1 & 0 & 0 & 0 & 0 \\
s^8 & s^7 & s^6 & s^5 & s^4 & s^3 & s^2 & s & 1 & 0 & 0 & 0 \\
s^9 & s^8 & s^7 & s^6 & s^5 & s^4 & s^3 & s^2 & s & 1 & 0 & 0 \\
s^{10} & s^9 & s^8 & s^7 & s^6 & s^5 & s^4 & s^3 & s^2 & s & 1 & 0 \\
s^{11} & s^{10} & s^9 & s^8 & s^7 & s^6 & s^5 & s^4 & s^3 & s^2 & s & 1 \\
s^{12} & s^{11} & s^{10} & s^9 & s^8 & s^7 & s^6 & s^5 & s^4 & s^3 & s^2 & s
\end{bmatrix}.$$

In[3]:=PartPoly[A]

$$\texttt{Out[3]=}\begin{bmatrix}
\frac{s}{1+s^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{1+s^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -s & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -s & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -s & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -s & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -s & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -s & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -s & \frac{1}{1+s^2} & \frac{s}{1+s^2}
\end{bmatrix}$$

Primer 2.28 Izborom test matrice F_{10} iz [86]:

$$F10 = \begin{bmatrix} 10+a & 9+a & 8+a & 7+a & 6+a & 5+a & 4+a & 3+a & 2+a & 1+a \\ 9+a & 9+a & 8+a & 7+a & 6+a & 5+a & 4+a & 3+a & 2+a & 1+a \\ 8+a & 8+a & 8+a & 7+a & 6+a & 5+a & 4+a & 3+a & 2+a & 1+a \\ 7+a & 7+a & 7+a & 7+a & 6+a & 5+a & 4+a & 3+a & 2+a & 1+a \\ 6+a & 6+a & 6+a & 6+a & 6+a & 5+a & 4+a & 3+a & 2+a & 1+a \\ 5+a & 5+a & 5+a & 5+a & 5+a & 5+a & 4+a & 3+a & 2+a & 1+a \\ 4+a & 4+a & 4+a & 4+a & 4+a & 4+a & 4+a & 3+a & 2+a & 1+a \\ 3+a & 3+a & 3+a & 3+a & 3+a & 3+a & 3+a & 3+a & 2+a & 1+a \\ 2+a & 2+a & 2+a & 2+a & 2+a & 2+a & 2+a & 2+a & 1+a & a \\ 1+a & 1+a & 1+a & 1+a & 1+a & 1+a & 1+a & 1+a & a & -1+a \end{bmatrix}.$$

In[4]:=PartPoly[F10]

$$\texttt{Out[4]=} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -\frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{5}{6} & \frac{7}{9}-\frac{a}{4} & \frac{4}{9} & \frac{1}{9}+\frac{a}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{4}{9} & \frac{1}{9} & -\frac{2}{9} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{9}+\frac{a}{4} & -\frac{2}{9} & -\frac{5}{9}-\frac{a}{4} \end{bmatrix}$$

Primer 2.29 Za test matricu označenu sa Z_{12} u [86]:

$$Z12 = \begin{bmatrix} a+1 & a & a & a & a & a & a & a & a & a & a & a \\ a & a-1 & a & a & a & a & a & a & a & a & a & a \\ a & a & a+1 & a & a & a & a & a & a & a & a & a \\ a & a & a & a-1 & a & a & a & a & a & a & a & a \\ a & a & a & a & a+1 & a & a & a & a & a & a & a \\ a & a & a & a & a & a-1 & a & a & a & a & a & a \\ a & a & a & a & a & a & a+1 & a & a & a & a & a \\ a & a & a & a & a & a & a & a-1 & a & a & a & a \\ a & a & a & a & a & a & a & a & a+1 & a & a & a \\ a & a & a & a & a & a & a & a & a & a-1 & a & a \\ a & a & a & a & a & a & a & a & a & a & a+1 & a \\ a & a & a & a & a & a & a & a & a & a & a & a-1 \end{bmatrix}.$$

Funkcija $PartPoly[Z12]$ daje poznati inverz Z_{12}^{-1}:

$$\texttt{Out[7]=} \begin{bmatrix} 1-a & a & -a & a & -a & a & -a & a & -a & a & -a & a \\ a & -1-a & a & -a & a & -a & a & -a & a & -a & a & -a \\ -a & a & 1-a & a & -a & a & -a & a & -a & a & -a & a \\ a & -a & a & -1-a & a & -a & a & -a & a & -a & a & -a \\ -a & a & -a & a & 1-a & a & -a & a & -a & a & -a & a \\ a & -a & a & -a & a & -1-a & a & -a & a & -a & a & -a \\ -a & a & -a & a & -a & a & 1-a & a & -a & a & -a & a \\ a & -a & a & -a & a & -a & a & -1-a & a & -a & a & -a \\ -a & a & -a & a & -a & a & -a & a & 1-a & a & -a & a \\ a & -a & a & -a & a & -a & a & -a & a & -1-a & a & -a \\ -a & a & -a & a & -a & a & -a & a & -a & a & 1-a & a \\ a & -a & a & -a & a & -a & a & -a & a & -a & a & -1-a \end{bmatrix}$$

Primer 2.30 Primer pozanat kao S_{11} iz [86]:

$$S11 = \begin{bmatrix} a+1 & a & a & a & a & a & a & a & a & a & a+1 \\ a & a-1 & a & a & a & a & a & a & a & a & a \\ a & a & a+1 & a & a & a & a & a & a & a & a \\ a & a & a & a-1 & a & a & a & a & a & a & a \\ a & a & a & a & a+1 & a & a & a & a & a & a \\ a & a & a & a & a & a-1 & a & a & a & a & a \\ a & a & a & a & a & a & a+1 & a & a & a & a \\ a & a & a & a & a & a & a & a-1 & a & a & a \\ a & a & a & a & a & a & a & a & a+1 & a & a \\ a & a & a & a & a & a & a & a & a & a-1 & a \\ a+1 & a & a & a & a & a & a & a & a & a & a+1 \end{bmatrix}.$$

Funkcijom $PartPoly[S11]$ dobijamo poznati inverz $S_{11}^{\dagger}$:

`Out[8]=`
$$\begin{bmatrix}
\frac{1-a}{4} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & \frac{1-a}{4} \\
\frac{a}{2} & -1-a & a & -a & a & -a & a & -a & a & -a & a \\
-\frac{a}{2} & a & 1-a & a & -a & a & -a & a & -a & a & -a \\
\frac{a}{2} & -a & a & -1-a & a & -a & a & -a & a & -a & a \\
-\frac{a}{2} & a & -a & a & 1-a & a & -a & a & -a & a & -a \\
\frac{a}{2} & -a & a & -a & a & -1-a & a & -a & a & -a & a \\
-\frac{a}{2} & a & -a & a & -a & a & 1-a & a & -a & a & -a \\
\frac{a}{2} & -a & a & -a & a & -a & a & -1-a & a & -a & a \\
-\frac{a}{2} & a & -a & a & -a & a & -a & a & 1-a & a & -a \\
\frac{a}{2} & -a & a & -a & a & -a & a & -a & a & -1-a & a \\
\frac{1-a}{4} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & -\frac{a}{2} & \frac{a}{2} & \frac{1-a}{4}
\end{bmatrix}$$

Primer 2.31 Neka je $A = \begin{bmatrix} s & 1 & 0 \\ s^2 & s & 1 \\ s^3 & s^2 & s \end{bmatrix}$.

`In[9]=PartPoly[A]`

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \frac{s}{1+s^2} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \frac{s}{1+s^2} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{s^2}{1+s^2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \frac{s^3}{1+s^2}$$

`Out[9]=`
$$\begin{bmatrix} \frac{s}{1+s^2} & 0 & 0 \\ \frac{1}{1+s^2} & 0 & 0 \\ \frac{-s-s^3}{1+s^2} & \frac{1}{1+s^2} & \frac{s}{1+s^2} \end{bmatrix}$$

2.2.2 Leverrier-Faddeev algoritam za polinomijalne matrice

Posmatrajmo polinomijalnu matricu $A(s) \in \mathbb{C}[s]^{n\times n}$. U [78] je predloženo sledeće tvrđenje.

Lema 2.3 [*78*] *Neka je data neregularna polinomijalna matrica $A(s)$ sa jednom promenljivom Predpostavimo da je*

$$\begin{aligned} a(z,s) &= \det\left[zI_n - A(s)\right] \\ &= a_0(s)z^n + a_1(s)z^{n-1} + \cdots a_{n-1}(s)z + a_n(s), \quad a_0(s) \equiv 1, \quad z \in \mathbb{C} \end{aligned}$$

karakterističan polinom za $A(s)$. Takođe, razmotrimo sledeći niz polinomijalnih matrica dimenzije $n \times n$

$$\begin{aligned} B_j(s) &= a_0(s)A(s)^j + a_1(s)A(s)^{j-1} + \cdots a_{j-1}(s)A(s) + a_j(s)I_n, \\ &\qquad a_0(s) = 1, \quad j = 0, \ldots, n \end{aligned}$$

Neka je

$$a_n(s) \equiv 0, \ldots, a_{t+1}(s) \equiv 0, \quad a_t(s) \neq 0.$$

Definišimo skup:

$$\Lambda = \{s_i \in \mathbb{C} : \ a_t(s_i) = 0\}.$$

Takođe, predpostavimo da

$$B_n(s) \equiv \mathbb{O}, \ldots, B_r(s) \equiv \mathbb{O}, \quad B_{r-1}(s) \neq \mathbb{O}$$

i $k = r - t$.

U slučaju $s \in \mathbb{C}\backslash\Lambda$ *i* $k > 0$, *Drazinov inverz za* $A(s)$ *dat je sa*

$$A(s)^D \quad = (-1)^{k+1} a_t(s)^{-k-1} A(s)^k B_{t-1}(s)^{k+1}.$$

U slučaju $s \in \mathbb{C}\backslash\Lambda$ *i* $k = 0$, *dobijamo* $A(s)^D = A(s)^{-1}$.

Za $s_i \in \Lambda$ *označimo sa* t_i *najveći ceo broj koji zadovoljava* $a_{t_i}(s_i) \neq 0$, *a sa* r_i *najmanji ceo broj koji zadovoljava* $B_{r_i}(s_i) \equiv \mathbb{O}$. *Onda je Drazinov inverz za* $A(s_i)$ *jednak*

$$A(s_i)^D = (-1)^{k_i+1} a_{t_i}(s_i)^{-k_i-1} A(s_i)^{k_i} B_{t_i-1}(s_i)^{k_i+1},$$

gde je $k_i = r_i - t_i$.

Saglasno rezultatu prethodne leme može se predložiti sledeći algoritam za izračunavanje Drazinovog inverza.

Algoritam 2.7 *Predpostavimo da je* $A(s) \in \mathbb{C}[s]^{n\times n}$ *polinomijalna matrica.*

Korak 1. *Konstruišimo niz brojeva* $\{a_0(s), a_1(s), \ldots, a_n(s)\}$ *i niz polinomijalnih matrica* $\{B_0(s), B_1(s), \ldots, B_n(s)\}$ *kao što sledi:*

$$\begin{array}{lll} A_0(s)=\mathbb{O}, & a_0(s)=1, & B_0(s)=I_n \\ A_1(s)=A(s)B_0(s), & a_1(s)=-\frac{Tr(A_1(s))}{1}, & B_1(s)=A_1(s)+a_1(s)I_n \\ \cdots & \cdots & \cdots \\ A_n(s)=A(s)B_{n-1}(s), & a_n(s)=-\frac{Tr(A_n(s))}{n}, & B_n(s)=A_n(s)+a_n(s)I_n. \end{array} \tag{2.2.13}$$

Korak 2. *Neka je*

$$t = \max\{l : a_l(s) \neq 0\}, \quad r = \min\{l : B_l(s) = \mathbb{O}\}, \quad k = r - t.$$

Za $s \in \mathbb{C}\backslash\Lambda$ *Drazinov inverz* $A(s)^D$ *dat je sa*

$$A(s)^D = (-1)^{k+1} a_t(s)^{-k-1} A(s)^k B_{t-1}(s)^{k+1} \tag{2.2.14}$$

Za one $s_i \in \Lambda$, *označimo sa* t_i *nakveći ceo broj koji zadovoljava* $a_{t_i}(s_i) \neq 0$, *a sa* r_i *najmanji ceo koji zadovoljava* $B_{r_i}(s_i) \equiv \mathbb{O}$. *Za celobrojne* $k_i = r_i - t_i$, *Drazinov inverz za* $A(s_i)$ *jednak je*

$$A(s_i)^D = (-1)^{k_i+1} a_{t_i}(s_i)^{-k_i-1} A(s_i)^{k_i} B_{t_i-1}(s_i)^{k_i+1}.$$

Predpostavimo sada da je $A(s) \in \mathbb{C}[s]^{n\times n}$ polinomijalna matrica u formi

$$A(s) = A_q s^q + A_{q-1} s^{q-1} + \cdots A_1 s + A_0 \in \mathbb{C}[s]^{n\times n}, \tag{2.2.15}$$

gde su $A_i \in \mathbb{C}^{n\times n}$, $i = 0,\dots,q$ konstantne kompleksne matrice a s nepoznata promenljiva. Koeficijent $a_i(s)$ može se posmatrati kao polinom od s.

Takođe, matrični koeficijenti $B_i(s)$ mogu se posmatrati kao polinomijalne matrice po stepenima promenljive s. Ovakav pristup daje analognu reprezentaciju Drazinovog inverza i analogni algoritam za izračunavanje Drazinovog inverza.

Teorema 2.23 *Neka je $A(s)$ polinomijalna matrica u formi* (2.2.15)*. Uočimo brojeve $\{a_0(s), a_1(s), \dots, a_n(s)\}$ i niz $\{B_0(s), B_1(s), \dots, B_n(s)\}$, polinomijalnih matrica definisani u Algoritmu 2.7. Napišimo $a_i(s)$ i $B_i(s)$ kao*

$$a_i(s) = \sum_{j=0}^{iq} a_{i,j}s^j, \quad i = 1,\dots,n, \tag{2.2.16}$$

$$B_i(s) = \sum_{j=0}^{iq} B_{i,j}s^j, \quad i = 1,\dots,n, \tag{2.2.17}$$

gde su $a_{i,j}$, $i=1,\dots,n$, $j=0,\dots,iq$ skalari a $B_{i,j}$, $i=1,\dots,n$, $j=0,\dots,iq$ konstantne matrice koje odgovaraju stepenima s^j. Tada je

$$a_{i+1,j} = -\frac{1}{i+1}\mathrm{Tr}\left(\sum_{\mathrm{l=0}}^{\mathrm{j}} \mathrm{A_{j-l}B_{i,l}}\right), \quad i = 0,1,\dots,n-1, \quad j = 0,1,\dots,(i+1)q, \tag{2.2.18}$$

$$B_{i+1,j} = \sum_{l=0}^{j} A_{j-l}B_{i,l} + a_{i+1,j}I_n, \quad i = 0,1,\dots,n-1, \quad j = 0,1,\dots,(i+1)q, \tag{2.2.19}$$

gde se u (2.2.18) *i* (2.2.19) *predpostava da je $A_k = \mathbb{O}$, $k \geq q+1$, $B_{ik} = \mathbb{O}$, $k \geq iq+1$.*

Dokaz. Nije teško pokazati da su iz (2.2.13) najveći stepen za $A_i(s)$ (a takođe i za $B_i(s)$) jednaki iq, $i = 0,\dots,n-1$. Takođe, stepen polinomnih veličina $a_i(s)$, $i = 1,\dots,n$ je većim delom jednak iq. Stoga se $a_i(s)$ i $B_i(s)$ kogu napisati u formi (2.2.16) i (2.2.17), respektivno. Takođe, iz (2.2.15), (2.2.13) i (2.2.17), dobijamo sledeće

$$\begin{aligned} A_{i+1}(s) &= A(s)B_i(s) = \left(\sum_{j=0}^{q} A_j s^j\right)\left(\sum_{l=0}^{iq} B_{i,l} s^l\right) \\ &= \sum_{j=0}^{(i+1)q}\left(\sum_{l=0}^{j} A_{j-l}B_{i,l}\right)s^j. \end{aligned} \tag{2.2.20}$$

Primena (2.2.13) i (2.2.20) vodi nas ka

$$\begin{aligned} a_{i+1}(s) &= -\frac{1}{i+1}\mathrm{Tr}(A_{i+1}(s)) \\ &= \sum_{j=0}^{(i+1)q}\left[-\frac{1}{i+1}\mathrm{Tr}\left(\sum_{\mathrm{l=0}}^{\mathrm{j}} \mathrm{A_{j-l}B_{i,l}}\right)\right]s^j. \end{aligned} \tag{2.2.21}$$

Jednakost (2.2.18) sledi iz (2.2.16) i (2.2.21).

Sa druge strane, koristeći (2.2.13), (2.2.21) i (2.2.22), dobijamo

$$\begin{aligned} B_{i+1}(s) &= A_{i+1}(s) + a_{i+1}(s)I_n \\ &= \sum_{j=0}^{(i+1)q}\left(\sum_{l=0}^{j} A_{j-l}B_{i,l}\right)s^j + \sum_{j=0}^{(i+1)q} a_{i+1,j}I_n s^j \\ &= \sum_{j=0}^{(i+1)q}\left(\sum_{l=0}^{j} A_{j-l}B_{i,l} + a_{i+1,j}I_n\right)s^j \end{aligned} \tag{2.2.22}$$

Jednakost (2.2.19) sledi iz (2.2.17) i (2.2.22). □

Sada smo u mogućnosti da uvedemo sledeći algoritam za izračunavanje Drazinovog inverza $A(s)^D$ za polinomijalnu matricu $A(s)$ u formi (2.2.15).

Algoritam 2.8 *Neka je*
Inicijalni postavke:

$$B_{0,0} = I_n, \quad A_k = 0, \;\; k = q+1, \dots, nq.$$

Ograničenja:

$$\begin{aligned} B_{0,j} &= \mathbb{O} \quad \forall j \in \mathbb{N}, \\ B_{i,j} &= \mathbb{O}, \quad i=0,\dots,n-1, \;\; j=iq+1,\dots,(n-1)q. \end{aligned}$$

Glavni algoritam:

$i = 0$

DO WHILE najmanje jedan $B_{i,j} <> 0$, $j = 0, 1, \dots, (i+1)q$:

Inicijalne predpostavke za B_{ij}:

$B_{i,j} = \mathbb{O}$, $j = iq+1, \dots, (n-1)q$.

Rekurzivna relacija za $a_i(s)$:

$$a_{i+1,j} = -\frac{1}{i+1}\mathrm{Tr}\left(\sum_{\mathrm{l=0}}^{\mathrm{j}} \mathrm{A_{j-l}B_{i,l}}\right), \ \mathrm{i=0,1,\dots,n-1,\ \ j=0,1,\dots,(i+1)q.}$$

Rekurzivna relacija za $B_i(s)$:

$$B_{i+1,j} = \sum_{l=0}^{j} A_{j-l}B_{i,l} + a_{i+1,j}I_n, \ \ i=0,1,\dots,n-1, \ \ j=0,1,\dots,(i+1)q.$$

$i = i+1$.

END DO.

Kriterijum zaustavljanja: Izračunati prvo t *koje zadovoljava*

$$a_{t+1,j} = a_{t+2,j} = \cdots = a_{n,j} = 0 \quad \forall j \in \mathbb{N}.$$

i prvo r *koje zadovoljava*

$$B_{r,j} = \mathbb{O}, \quad j = 0, 1, \dots, rq.$$

Izlaz: Izračunati $k = r - t$ *i polinomijalnu matricu*

$$A(s)^D = (-1)^{k+1}\left(\sum_{j=0}^{tq} a_{r-1,j}s^j\right)^{-k-1}\left(\sum_{i=0}^{q} A_i s^i\right)^k\left(\sum_{l=0}^{(t-1)q} B_{t-1,l}s^l\right)^{k+1} \tag{2.2.23}$$

Primer 2.32 Neka je data polinomijalna matrica

$$A(s) = \begin{bmatrix} 1+s & s & 1+s \\ s & -1+s & s \\ 1+s & s & 1+s \end{bmatrix}.$$

Primenom *Algoritma 2.8* imamo

$$\begin{aligned}
A_0(s) &= 0_{3,3};\ a_0(s) = 1;\ B_0(s) = I_3; \\
A_1(s) &= A(s)B_0(s) = A(s)\ \ a_1(s) = -\frac{\mathrm{Tr}(A_1(s))}{1!} = -(1+3s); \\
B_1(s) &= A_1(s) + a_1(s)I_3 = \begin{bmatrix} -2s & s & 1+s \\ s & -2-2s & s \\ 1+s & s & -2s \end{bmatrix};
\end{aligned}$$

$$\begin{aligned}
A_2(s) &= A(s)B_1(s) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix};\\
a_2(s) &= -\frac{\mathrm{Tr}(A_2(s))}{2!} = -\frac{4}{2} = -2;\\
B_2(s) &= A_2(s) + a_2(s)I_3 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix};\\
A_3(s) &= A(s)B_2(s) = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} = 0_{3,3}; \quad a_3(s) = -\frac{\mathrm{Tr}(A_3(s))}{3!} = 0.
\end{aligned}$$

Prema tome

$$t = \max\{l : a_l(s) \neq 0\} = 2, \quad r = \min\{l : B_l(s) = \mathbb{O} = 3, \quad k = r - t = 1.$$

Tim pre

$$\begin{aligned}
A(s)^D &= a_2(s)^{-1-1}A(s)^1B_1(s)^{1+1}\\
&= (-2)^{-2}\begin{bmatrix} 1+s & s & 1+s \\ s & -1+s & s \\ 1+s & s & 1+s \end{bmatrix}\begin{bmatrix} -2s & s & 1+s \\ s & -2-2s & s \\ 1+s & s & -2s \end{bmatrix}^2\\
&= \begin{bmatrix} \frac{1}{4}-\frac{1}{4}s & \frac{1}{2}s & \frac{1}{4}-\frac{1}{4}s \\ \frac{1}{2}s & -1-s & \frac{1}{2}s \\ \frac{1}{4}-\frac{1}{4}s & \frac{1}{2}s & \frac{1}{4}-\frac{1}{4}s \end{bmatrix}.
\end{aligned}$$

Rezultat standardne funkcije *Tr[g_]* je trag matrice g. Polinomijalna forma matrice (2.2.15) reprezentovana je tro-dimenzionalnom listom $\{l_1, \dots, l_{q+1}\}$, gde elementi l_i označavaju matrice A_{i-1}, $i = 1, \dots, q+1$. Elementi globalnog niza A jednaki su

$$A[[i]] = l_{i-1}, \;\; i = 0, \dots, q, \quad A[[i]] = \mathbb{O}, \;\; i = q+1, \dots, nq.$$

Rekurzivne realacije (2.2.18) i (2.2.19) za izračunavanje realnog broja $a_{i,j}$ i matrice $B_{i,j}$, respektivno, implementirani su sledećim dvema funkcijama.

```
Fa[i1_,j1_]:= -1/i1*Tr[Sum[A[[j1-l+1]].FB[i1-1,l],l,0,j1]]

FB[u_, v_]:= Block[{w={}},
    nul=Table[0, {n}, {n}];
    If[u==0 && v==0, w=IdentityMatrix[n],
    If[u==0, w=nul,
      If[(v>=u q+1) && (v<=(n-1) q), w=nul,
```

```
        w=Sum[A[[v-k+1]].FB[u-1,k], {k,0,v}]+Fa[u,v]*IdentityMatrix[n]]
    ] ];
     Return[w]
];
```

Shodno pomoćnoj funkciji *FrmA[a_List,var_]* moguće je transformisati polinomijalnu matricu $A[var]$ u internu reprezentaciju u formi $\{l_1, \ldots, l_{q+1}\}$, gde lista l_i predstavlja matrice A_{i-1}, $i = 1, \ldots, q+1$.

```
FrmA[a_List,var_]:=
   Block[{L={},L1,nul,n,n1,m1,i},
     {n1,m1}=Dimensions[a]; nul=Table[0,{i,n1},{j,m1}]; L1=nul;
      n=Max[Exponent[a,var,List]];
      For[i=1,i<=n,i++,
         koef=Coefficient[a,var^i];
         L=Append[L,koef];      L1=L1+koef*var^i
      ];
      If[L==={}, Return[{a-L1}] ];
      L1={a-L1};
      For[i=1,i<=Length[L],i++, AppendTo[L1,L[[i]]]; ];
      Return[L1]
   ]
```

Metod za izračunavanje Drazinovog inverza koji je opisan u *Algoritmu 2.8*, implementiran je sledećom funkcijom *DrzPoly[mat_List]*, gde se pretpostavlja da formalni parametri *mat_* označavaju tro-dimenzionalnu listu u formi $\{A_0, \ldots, A_q\}$.

```
DrzPoly[mat_List]:=
   Block[{A,i,j,nul,k,q,r=t=0,log1=log2=True,var=Variables[mat]},
     If[var=={}, A={mat}, A=FrmA[mat,First[var]]];
     q=Length[A]-1; {n,n}=Dimensions[A[[1]]]; nul=Table[0,{n},{n}];
     For[i=q+1, i<=n q, i++, A=Append[A, nul]];
     For[i=0, i<=n-1, i++,
       For[j=0, j<=(i+1) q, j++,
          If[Fa[i+1,j]==0, log1=log1 && True, log1=log1 && False];
         If[FB[i+1,j]===nul, log2=log2 && True, log2=log2 && False];
       ];
       If[!log1, t=i+1; log1=True];   If[!log2, r=i+1];
     ];
     k=r-t;
     rez1=(-1)^(k+1)*Sum[Fa[t,j]*First[var]^j, {j,0,t q}]^(-k-1);
     rez2=MatrixPower[Sum[A[[i+1]] First[var]^i, {i,0,q}],k];
     rez3=MatrixPower[Sum[FB[t-1,l] First[var]^l, {l,0,(t-1) q}], k+1];
```

```
    rez = rez1*rez2.rez3;
    Return[Simplify[rez]]
  ]
```

Primer 2.33 Polinomijalna matrica

$$A(s)=\begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix}+\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}s+\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}s^2=\begin{bmatrix} 1+s & s & 1+s \\ s^2 & -1+s & s \\ 1+s & s & 1+s \end{bmatrix}$$

reprezentovana je sledećom trodimenzionalnom listom, generiasana izrazom $mat = FrmA[a,s]$:

$\{\{\{1,0,1\},\{0,-1,0\},\{1,0,1\}\},\{\{1,1,1\},\{0,1,1\},\{1,1,1\}\},\{\{0,0,0\},\{1,0,0\},\{0,0,0\}\}\}$.

Primenom programa $DrzPoly[mat]$, dobijamo da je $t = 2$, $r = 3$ i Drazin inverz za $A(s)$:

$$A(s)^D = \begin{bmatrix} \frac{1-s+2s^3-2s^4}{(2-s^2+s^3)^2} & \frac{s}{2-s^2+s^3} & \frac{1-s-s^2+s^4}{(2-s^2+s^3)^2} \\ \frac{s(-1+s^2+s^3)}{(1+s)(2-2s+s^2)^2} & -\frac{2}{2-2s+s^2} & \frac{3s-2s^3}{(1+s)(2-2s+s^2)^2} \\ \frac{1-s+2s^3-2s^4}{(2-s^2+s^3)^2} & \frac{s}{2-s^2+s^3} & \frac{1-s-s^2+s^4}{(2-s^2+s^3)^2} \end{bmatrix}.$$

Napomenimo da su sledeće vrednosti dobijene za koeficijente $a_{i,j}$:

A[1,0]=-1; A[1,1]=-3; A[1,2]=0; A[2,0]=-2; A[2,1]=0; A[2,2]=1; A[2,3]=-1;
A[2,4]=0; A[3,0]=A[3,1]=$\cdots$=A[3,6]=0.

a sledeće za vrednsti matrica B_{ij}:

B[1,0]=$\{\{0,0,1\},\{0,-2,0\},\{1,0,0\}\}$; B[1,1]=$\{\{-2,1,1\},\{0,-2,1\},\{1,1,-2\}\}$;
B[1,2]=$\{\{0,0,0\},\{1,0,0\},\{0,0,0\}\}$; B[2,0]=$\{\{-1,0,1\},\{0,0,0\},\{1,0,-1\}\}$;
B[2,1]=$\{\{0,0,0\},\{1,0,-1\},\{0,0,0\}\}$; B[2,2]=$\{\{0,0,0\},\{0,0,0\},\{-1,0,1\}\}$;
B[2,3]=$\{\{0,0,0\},\{-1,0,1\},\{1,0,-1\}\}$; B[2,4]=$\{\{0,0,0\},\{-1,0,1\},\{1,0,-1\}\}$;
B[3,0]=B[3,1]=$\cdots$=B[3,6]=$\{\{0,0,0\},\{0,0,0\},\{0,0,0\}\}$.

2.2.3 Efektivni algoritam Leverrier-Faddeevog tipa

Lako se proverava da se broj koeficijenata a_{ij} i matrica $B_{i,j}$ ugrađenih u izračunavanju Drazinovog inverza $A(s)^D$ za $A(s) \in A[s]^{n\times n}$ jednak

$$\sum_{i=1}^{n}(iq+1) = \frac{n(n+1)}{2}q + n.$$

Stoga, čak i u slučaju kad je uz s samo jedan najveći stepen u $A(s)$, npr. s^{80}, broj matrica koje se koriste za izračunavanje Drazinovog inverza za $A(s)$ je $\frac{n(n+1)}{2}80 + n$, iako su odsutne matrice stepena $s^{79}, \ldots, s$. U tom slučaju kad postoji velike praznine između stepena za s u $A(s)$, predložen je niz koraka za poboljšanje algoritma.

Definišimo sa:

$$\begin{aligned}\Phi_A &= \{(\mu_i) : \text{skup stepena nenula koeficijenata matrice A(s)}\},\\ \Phi_A(i) &= i\text{-ti element za } \Phi_\text{A} \text{ (neka je } (\mu_\text{i})),\\ n_A &= q = \text{ukupan broj elemenata u } \Phi_\text{A}.\end{aligned}$$

Onda se polinomjalna matrica $A(s) \in \mathbb{C}[s]^{n\times n}$ može predstaviti u formi

$$A(s) = \sum_{i=1}^{q} A_{\Phi_A(i)} s^{\Phi_A(i)} = \sum_{i=1}^{q} A_{\mu_i} s^{\mu_i}, \tag{2.2.24}$$

gde su $A_{\Phi_A(i)} = A_{\mu_i} \in \mathbb{C}^{n\times n}$, $i = 0,\ldots,q$ konstantne nenula kompleksne matrice u (2.2.15)). Takođe, napišimo $a_i(s)$ i $B_i(s)$ kao

$$a_i(s) = \sum_{j=1}^{n_i} a_{i,\Phi_i(j)} s^{\Phi_i(j)} \tag{2.2.25}$$

$$B_i(s) = \sum_{j=1}^{n_i} B_{i,\Phi_i(j)} s^{\Phi_i(j)}, \tag{2.2.26}$$

gde su:

$$\begin{aligned}\Phi_i &= \text{skup stepena nenula koeficijenata matrice } B_i(s),\\ \Phi_i(j) &= j\text{-ti element za } \Phi_i,\\ n_i &= [\Phi_i] = \text{ukupan broj elemenata u}\, \Phi_i.\end{aligned}$$

Onda je

$$\begin{aligned}A_{i+1}(s) &= A(s)B_i(s) = \left(\sum_{j=1}^{n_A} A_{\Phi_A(j)} s^{\Phi_A(j)}\right)\left(\sum_{k=1}^{n_i} B_{i,\Phi_i(k)} s^{\Phi_i(k)}\right)\\ &= \sum_{j=1}^{n_A}\sum_{k=1}^{n_i} A_{\Phi_A(j)} B_{i,\Phi_i(k)} s^{\Phi_A(j)+\Phi_i(k)} = \sum_{j=1}^{n_A}\sum_{k=1}^{n_i} Q_{\Phi_A(j)+\Phi_i(k)} s^{\Phi_A(j)+\Phi_i(k)}.\end{aligned}$$

S'obzorom da je,

$$\Phi_{i+1} = \Phi_{i+1} \cup \{\Phi_A(j) + \Phi_i(k)\},\ i=1,\ldots,n-1,\ j=1,\ldots,n_A,\ k=1,\ldots,n_i.$$

Izbacimo iz skup Φ_{i+1}, $i = 1,\ldots,n-1$ one stepene $\Phi_{i+1}(j)$ koji odgovaraju nula proizvodima $A_{\Phi_A(j)} B_{i,\Phi_i(k)}$ i formitrajmo novi skup Φ_{i+1} sa ukupnim brojem elemenata n_{i+1} umesto $\tilde{n}_{i+1}$ za Φ_{i+1} koji su predhodili.

Zamenom (2.2.24), (2.2.25) i (2.2.26) u rekurzivnim relacijama (2.2.13) dobija se rekurzivni algoritam koji određuje $a_{i+1,\Phi_{i+1}(j)}$ i $B_{i+1,\Phi_{i+1}(j)}$ za $j = 1,\ldots,n_i$.

Algoritam 2.9 *(Izračunavanje Drazinovog inverza za $A(s)$).*
Inicijalizacija:

$$B_{0,0} = I_n.$$

Granični uslovi:

$$\begin{aligned}
\Phi_i &= \{0\}, \quad n_i = 1, \quad i = 0, 1, \dots, n, \\
n_E &= \text{maksimalni eksponent za s u A(s)}, \\
Q_i &= \mathbb{O}, \quad i = 0, 1, \dots, 2n_E + 1, \\
\Phi_A &= \{\Phi_A(1), \dots, \Phi_A(n_A)\} = \text{skup nenula koeficijenata matrica za A(s)}, \\
n_A &= q = \text{ukupan broj elemenata u } \Phi_\mathrm{A}.
\end{aligned}$$

Glavni program:
$i = 0$
DO WHILE $\Phi_i \neq \emptyset \wedge i < n$.
Korak 1. Izračunati:
a) koeficijente matrica koji odgovaraju za $\Phi_A(j) + \Phi_i(k)$ *- stepene s u* $A(s)B_i(s)$,
b) skup Φ_{i+1} *u terminima* Φ_A *i* Φ_i

$$\begin{aligned}
Q_{\Phi_A(j)+\Phi_i(k)} &= Q_{\Phi_A(j)+\Phi_i(k)} + A_{\Phi_A(j)} B_{i,\Phi_i(k)}, \\
\Phi_{i+1} &= \Phi_{i+1} \cup \{\Phi_A(j) + \Phi_i(k)\}, j = 1, \dots, n_A, \ k = 1, \dots, n_i.
\end{aligned}$$

Korak 2. Izračunati ukupan broj elemenata u Φ_{i+1}

$$\tilde{n}_{i+1} = \text{ukupan broj elemenata u } \Phi_{\mathrm{i+1}}$$

Korak 3. Izračunati $a_{i+1,\Phi_{i+1}(j)}$ *i* $B_{i+1,\Phi_{i+1}(j)}$*:*
Postaviti $s = 0$
Za $j = 1, \dots, \tilde{n}_{i+1}$ *obaviti sledeće:*
Ako je $Q_{\Phi_{i+1}(j)} = \mathbb{O}$ *onda izvršiti:*

$$\Phi_{i+1} = \Phi_{i+1} - \{\Phi_{i+1}(j)\}, \quad s = s + 1,$$

inače izračunati

$$a_{i+1,\Phi_{i+1}(j)} = -\frac{1}{i+1} \mathrm{Tr}\left(Q_{\Phi_{i+1}(j)}\right).$$

kraj uslova.

Ako je $i < n-1$ onda

$$B_{i+1,\Phi_{i+1}(j)} = Q_{\Phi_{i+1}(j)} + a_{i+1,\Phi_{i+1}(j)} I_r$$

kraj uslova.

Postaviti $Q_{\Phi_{i+1}(j)} = \mathbb{O}$

Uzeti sledeće j

$n_{i+1} = \tilde{n}_{i+1} - s$

$i = i + 1$

END DO

Kriterijum prekida:

Naći $t = \max\{l : \ a_{l,\Phi_l(j)} \neq 0\}$ *i* $r = \min\{l : \ B_l(s) = \mathbb{O}\}$, *kao i naći*

$$t: \ a_{t+1,\Phi_{t+1}(j)} = a_{t+2,\Phi_{t+2}(j)} = \cdots = a_{n_i,\Phi_{n_i}(j)} = 0,$$
$$r: \ B_{r,\Phi_r(i)} = 0, \ i = 1, \ldots, n_i.$$

Za $k = r - t$ *postaviti*

$$\begin{aligned} B_{\Phi_{t-1}(j)} &= B_{t-1,\Phi_{t-1}(j)}, \quad j = 1, \ldots, n_{t-1}, \\ a_{\Phi_t(j)} &= a_{t,\Phi_t(j)}, \quad j = 1, \ldots, n_t. \end{aligned}$$

Izlaz:

$$\begin{aligned} A(s)^D &= (-1)^{k+1} \left(\sum_{j=1}^{n_t} a_{\Phi_t(j)} s^{\Phi_t(j)} \right)^{-k-1} \left(\sum_{i=1}^{q} A_{\Phi_A(i)} s^{\Phi_A(i)} \right)^k \times \\ &\quad \times \left(\sum_{l=1}^{n_{t-1}} B_{\Phi_{t-1}(l)} s^{\Phi_{t-1}(l)} \right)^{k+1}. \end{aligned}$$

Napomena 2.4 *Broj potrebnih izračunavanja koeficijenata* $a_{i,\Phi_i(j)}$ *i koeficijenata matrice* $B_{i,\Phi_i(j)}$ *jednak je* $\sum_{j=1}^{q} n_i$, *koje zavise od* n *i* q.

Primer 2.34 Neka je polinomijalna matrica $A(s)$ data kao:

$$A(s) = \begin{bmatrix} 1+s & s & 1+s \\ s & -1+s & s \\ 1+s & s & 1+s \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} s + \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = A_1 s + A_0.$$

Primenom Algoritma 2.9 dobijamo:

Inicijalno:

$$B_{0,0} = I_n = I_3.$$

Granični uslovi:

$$\begin{aligned}
\Phi_i &= \{0\},\ n_i = 1, \mathrm{za} i = 0,1,2,3,\\
\Phi_A &= \{0,1\}, \quad \Phi_A(1) = 0,\ \Phi_A(2) = 1,\\
n_A &= 2 = \text{ukupni broj elemenatat u } \Phi_\mathrm{A},\\
Q_i &= \mathbb{O}_{3,3}, \quad i = 0,1.
\end{aligned}$$

Korak 1. Primenimo za $i = 0,1,2$ sledeće korake.
$i = 0$
Korak 1.1.

$$\begin{aligned}
Q_0 &= Q_0 + A_0 B_{0,0} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (j=1,\ k=1);\\
Q_1 &= Q_1 + A_1 B_{0,0} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (j=2,\ k=1);
\end{aligned}$$

$$\Phi_1 = \Phi_1 \cup \{0,1\} = \{0,1\}.$$

Korak 1.2. Izračunati ukupan broj elemenata u Φ_1

$$\tilde{n}_1 = 2 = \text{ukupan broj u } \Phi_1.$$

Korak 1.3. Izračunati $a_{1,\Phi_1(j)}$ i $B_{1,\Phi_1(j)}$:
Postaviti $s = 0$

$$\begin{aligned}
Q_0 \neq \mathbb{O}_{3,3} \quad &\Rightarrow \quad a_{1,0} = -\frac{1}{1}\mathrm{Tr}(Q_0) = -1 \quad (j=1);\\
B_{1,0} \quad &= \quad Q_0 + a_{1,0} I_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix};\\
Q_1 \neq \mathbb{O}_{3,3} \quad \Rightarrow a_{1,1} &= -\frac{1}{1}\mathrm{Tr}(Q_1) = -3 \quad (j=1);\\
B_{1,1} \quad &= \quad Q_1 + a_{1,1} I_3 = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix};
\end{aligned}$$

$$n_1 = \tilde{n}_1 - s = 2 - 0 = 2.$$

Izbaciti ulaze Q_i:

$$Q_0 = Q_1 = \mathbb{O}_{3,3}$$

$i = 1.$

Korak 1.1.

$$\begin{aligned}
Q_0 &= Q_0 + A_0 B_{1,0} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (j=1,\ k=1); \\
Q_1 &= Q_1 + A_0 B_{1,0} = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad (j=1,\ k=2); \\
Q_1 &= Q_1 + A_1 B_{1,0} = \mathbb{O}_{3,3} \quad (j=2,\ k=1); \\
Q_2 &= Q_2 + A_1 B_{1,1} = \mathbb{O}_{3,3} \quad (j=2,\ k=2);
\end{aligned}$$

$$\Phi_2 = \Phi_2 \cup \{0,1,2\} = \{0,1,2\}.$$

Korak 1.2. Izračunati ukupan broj elemenata u Φ_2.

$$\tilde{n}_2 = 3 = \text{ukupan broj elemenata u } \Phi_2$$

Korak 1.3. Izračunati $a_{2,\Phi_2(j)}$ i $B_{2,\Phi_2(j)}$:
Postaviti $s = 0$

$$\begin{aligned}
Q_0 \neq \mathbb{O}_{3,3} \quad &\Rightarrow \quad a_{2,0} = -\frac{1}{2}\mathrm{Tr}(Q_0) = -2 \quad (j=1); \\
B_{2,0} \quad &= \quad Q_0 + a_{2,0} I_3 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}; \\
Q_1 = \mathbb{O}_{3,3} \quad &\Rightarrow \quad \Phi_2 = \Phi_2 - \{1\} = \{0,2\},\ s{=}s+1{=}1 \quad (j=2); \\
Q_2 = \mathbb{O}_{3,3} \quad &\Rightarrow \quad \Phi_2 = \Phi_2 - \{2\} = \{0\},\ s{=}s+1{=}2 \quad (j=3);
\end{aligned}$$

$$n_2 = \tilde{n}_2 - s = 3 - 2 = 1.$$

Izbaciti ulaze Q_i:

$$Q_0 = \mathbb{O}_{3,3}.$$

$i = 2.$

Korak 1.1.

$$\begin{aligned}
Q_0 &= Q_0 + A_0 B_{2,0} = \mathbb{O}_{3,3} \quad (j=1,\ k=1); \\
Q_1 &= Q_1 + A_1 B_{1,0} = \mathbb{O}_{3,3} \quad (j=2,\ k=1);
\end{aligned}$$

$$\Phi_3 = \Phi_3 \cup \{0,1\} = \{0,1\}.$$

Korak 1.2. Izračunati ukupan broj elemenata u Φ_3

$$\tilde{n}_3 = 2 = \text{ukupan broj elemenata u } \Phi_3$$

Korak 1.3. Iztračunati $a_{3,\Phi_3(j)}$ i $B_{3,\Phi_3(j)}$:
Postaviti $s = 0$

$$\begin{aligned} Q_0 &= \mathbb{O}_{3,3} \quad \Rightarrow \quad \Phi_3 = \Phi_3 - \{0\} = \{1\},\ s{=}s+1{=}1 \quad (j=1); \\ Q_1 &= \mathbb{O}_{3,3} \quad \Rightarrow \quad \Phi_3 = \Phi_3 - \{1\} = \emptyset,\ s{=}s+1{=}2 \quad (j=2); \end{aligned}$$

$$n_3 = \tilde{n}_3 - s = 2 - 2 = 0.$$

$i = 3$.

$$\Phi_3 = \emptyset \text{ END DO.}$$

Kriterijum zaustavljanja:

$$t = 2 = \max\{l, a_{l,\Phi_l(j)} \neq 0\}, \quad r = i = 3 = \min\{l : B_l(s) = \mathbb{O}\}.$$

Postaviti

$$B_0 = B_{1,0}, \quad B_1 = B_{1,1} \quad a_0 = a_{2,0}$$

IZLAZ: Za $k = r - t = 3 - 2 = 1$ Drazinov inverz za $A(s)$ je

$$\begin{aligned} A(s)^D &= (-1)^2 \tfrac{A(s)^1 B_{2-1}(s)^{1+1}}{a_2(s)} = \tfrac{(A_0+A_1 s)^1 (B_0+B_1 s)^2}{a_2(s)} \\ &= \begin{bmatrix} -\frac{1}{4}s + \frac{1}{4} & \frac{1}{2}s & -\frac{1}{4}s + \frac{1}{4} \\ \frac{1}{2}s & -1 - s & \frac{1}{2}s \\ -\frac{1}{4}s + \frac{1}{4} & \frac{1}{2}s & -\frac{1}{4}s + \frac{1}{4} \end{bmatrix}. \end{aligned}$$

Metod za izračunavanje Drazinovog inverza opisan u *Algoritamu 2.9*, implementiran je sledećom funkcijom $DrzEf[mat\_List]$, gde se predpostavlja da formalni parametar $mat\_$ predstavlja tro-dimenzionalnu listu

$$A1 = FrmA[mat, First[Variables[mat]]]$$

koja sadrži konstantne koeficijentne matrice $A_0, \ldots, A_q$ iz (2.2.15).

```
DrzEf[mat_List]:=
   Module[{A1,i,j,ne,na,s,r=t=0,log1=log2=True,
    var=Variables[mat],FA,FK,m,n},
    If[var=={}, A1=mat, A1=FrmA[mat, First[var]]];
    {n,m}=Dimensions[mat]; nul=Table[0, {i,n}, {j,n}];
    FA={}; ne=Max[Exponent[mat, First[var]]];
    For[i=0, i<Length[A1], i++,
```

```
  If[A1[[i+1]]=!=nul, AppendTo[FA, i]; A[i]=A1[[i+1]];
] ];
na=Length[FA];
For[i=0, i<=2*ne+1, i++, Q[i]=nul];
For[i=0, i<=n, i++, F[i]={0}; ni[i] = 1];
i=0; B[0,0]=IdentityMatrix[n];
While[(F[i]=!={}) && (i<n),
  For[j=1, j<=Length[FA], j++,
    For[k = 1, k <= ni[i], k++,
      Q[FA[[j]]+F[i][[k]]]=Q[FA[[j]]+F[i][[k]]]
                    +A[FA[[j]]].B[i,F[i][[k]]];
      F[i+1]=Union[F[i+1], {FA[[j]]+F[i][[k]]}];
  ] ];
  nk[i+1]=Length[F[i+1]];  s=0; FK={};
  For[j=1, j<=nk[i+1], j++,
    If[Q[F[i+1][[j]]]===nul,
      FK=Append[FK,F[i+1][[j]]];   s=s+1,
      a[i+1, F[i+1][[j]]]=-1/(i+1)*Tr[Q[F[i+1][[j]]]];
      If[a[i+1, F[i+1][[j]]]===0,
          log1=log1 && True, log1=log1 && False];
      If[i<(n-1),
        B[i+1,F[i+1][[j]]]=Q[F[i+1][[j]]]
               +a[i+1,F[i+1][[j]]]*IdentityMatrix[n];
        If[B[i+1,F[i+1][[j]]]===nul,
           log2=log2 && True, log2=log2 && False];
      ];
      Q[F[i+1][[j]]]=nul;
  ] ];
  F[i+1]=Complement[F[i+1],FK];
  If[log1, t=i, log1=True]; If[log2, r=i+1, log2=True];
  ni[i+1]=nk[i+1]-s;       i=i+1;
];
k=r-t;
If[t==0, Return[Simplify[Inverse[mat]]]];
rez1=(-1)^(k+1)*Sum[a[t,F[t][[j]]]*First[var]^F[t][[j]],
                                   {j,1,ni[t]}]^(-k-1);
rez2=MatrixPower[Sum[A[FA[[i]]]*First[var]^FA[[i]],
                                   {i,1,na}], k];
rez3=MatrixPower[Sum[B[t-1,F[t-1][[l]]]*First[var]^F[t-1][[l]],
                                  {l,1,ni[t-1]}], k+1];
Clear[Q]; Clear[B]; Clear[ni];  Clear[nk]; Clear[F]; Clear[a];
```

```
 Return[MatrixForm[Simplify[rez1*rez2.rez3]]];
]
```

Primer 2.35 U ovom primeru pokazana je efikasnost Algoritma 2.9 u odnosu na Algoritam 2.8. Posmatrajmo matricu

$$B(s) = \begin{bmatrix} 1+s & s & 1+s \\ s^q & -1+s & s \\ 1+s & s & 1+s \end{bmatrix}$$

za različite vrednosti parametra q.

q	DrzPoly[B(s)]	DrzEf[B(s)]	Timing[DrzPoly[B(s)]]	Timing[DrzEf[B(s)]]
1	9	3	$0.06 Second$	$0.01 Second$
2	15	7	$0.27 Second$	$0.01 Second$
3	21	8	$0.751 Second$	$0.01 Second$
4	27	8	$1.072 Second$	$0.01 Second$
5	33	8	$3.284 Second$	$0.01 Second$
6	39	8	$5.809 Second$	$0.01 Second$
7	45	8	$9.524 Second$	$0.01 Second$
8	51	8	$14.851 Second$	$0.01 Second$
9	57	8	$22.052 Second$	$0.01 Second$
10	63	8	$31.475 Second$	$0.01 Second$
11	69	8	$43.732 Second$	$0.01 Second$
12	75	8	$59.286 Second$	$0.01 Second$
13	81	8	$78.563 Second$	$0.01 Second$
14	87	8	$102.397 Second$	$0.01 Second$
15	93	8	$131.209 Second$	$0.01 Second$
16	99	8	$165.457 Second$	$0.01 Second$
17	105	8	$206.117 Second$	$0.01 Second$
18	111	8	$253.504 Second$	$0.01 Second$
19	117	8	$310.156 Second$	$0.01 Second$
20	123	8	$374.188 Second$	$0.01 Second$
25	151	8	$857.342 Second$	$0.01 Second$
30	183	8	$1695.46 Second$	$0.01 Second$
35	213	8	$3044.13 Second$	$0.01 Second$
40	283	8	$-$	$0.02 Second$
80	483	8	$-$	$0.03 Second$

U Tabeli 1 pokazana je efikasnost u odnosu na CPU vreme i broj poterbnih vrednosti za računanje. Broj vrednosti $a_{i,j}$ i matrica $B_{i,j}$, za funkcijama $DrzPoly[B(s)]$ i $DrzEf[B(s)]$, upisane su u 2 i 3 koloni Tabele 1. Takođe, CPU vreme je prezentovano u 4 i 5 koloni, respektivno. Ovi rezultati dobijeni su pomoću paketa MATHEMATICA verzije 4.1 i Pentium-a III 1.2 GHz.

Napomenimo da crta (-) u tabeli označava veoma dugo vreme. Možemo primetiti da je čak i za $q = 1$ broj potrebnih vrednosti $a_{i,j}$ i matrica $B_{i,j}$, koje koriste funkcija $DrzPoly[B(s)]$ i $DrzEf[B(s)]$ jednaka 9 i 3, respektivno.

Otuda i zaključak da *Algoritam 2.9* zauzima manje memorijskog prostora čak i u slučaju kad ne postoji razlike u stepenima polinomijalnih matrica. Stoga je, efektivni algoritam bolje koristiti nego *Algoritam 2.8.*

2.2.4 Inverz polinomialnih matrica interpolacijom

Poznato je da postoji jedan i samo jedan polinom stepena $q \leq n$ sa vrednostima $f(s_0), f(s_1), \ldots, f(s_n)$ u različitim tačkama $s_0, s_1, \ldots, s_n$. Polinom se naziva interpolacioni polinom $q$tog stepena. Najvažnija su tri interpolaciona methoda [10]:

i) direktan pristup kji koristi Vandermondeove matrice;
ii) Newtonova interpolacija;
iii) Lagrangeova interpolacija.

Generalizujući ove interpolacione metode na slučaj polinomijalnih matrica $A(s)$, algoritmi za inverziju polinomijalnih matrica mogu se razviti kao numerički stabilni algoritmi, koji manipulišu nad konstantnim matricama.

Između ova tri metoda naročito je pogodno koristiti Lagrange-ovu interpolaciju.

Sada ćemo uvesti algoritam za izračunavanje inverza polinomijalnih matrica inverzijom.

Algoritam 2.10 *Neka je data matrica $A(s)$ sa nepoznatim parametrom s.*

Korak 1. Izabrati različite početne tačke interpolacije: $s_0, s_1, \ldots, s_n$, i $s_i \in \mathbb{R}$.

Za svako $i = 0, 1, \ldots, n$:

Korak 2. Naći konstantne matrice $A_i = A(s_i)$, u svakoj početnoj tački s_i;

Korak 3. Izračunati Moore-Penrose inverz $A_i^\dagger$ za sve konstantne matrice A_i;

Korak 4. Moore-Penrose inverz $A(s)^\dagger$ date matrice $A(s)$ jednak je

$$A(s)^\dagger = \sum_{i=0}^{n} A_i^\dagger L_i(s), \tag{2.2.27}$$

gde je $L_i(s) = \frac{\prod\limits_{\substack{k=0\\k\neq i}}^{n}(s-s_k)}{\prod\limits_{\substack{k=0\\k\neq i}}^{n}(s_i-s_k)}$.

U proceduralnim programskim jezicima bez mogućnosti simboličkog izračunavanja neophodno je datu matricu $A(s)$ dimenzije $m \times n$, sa nepoznatim parametrom s, transformisati u polinomialnu formu:

$$A[s] = B_0 + B_1 s + \cdots + B_{q-1} s^{q-1} + B_q s^q = \sum_{i=0}^{q} B_i s^i \qquad (2.2.28)$$

gde su B_i, $i = 0, \ldots, q$, $q \leq n$, kostantne matrice dimenzije $m \times n$.

U tom slučaju u *Koraku 4.* moraju se naći koeficijenti interpolacionog polinoma $L_i(s)$.

Transformacijom $L_i(s)$ u polinomijalnu formu

$$L_i(s) = l_{i,0} + l_{i,1} s + l_{i,2} s^2 + \cdots + l_{i,q} s^q, \quad i = 0, 1, \ldots, n$$

i

$$A(s)^\dagger = \sum_{i=0}^{n} A_i^\dagger L_i(s) = \sum_{i=0}^{n} A_i^\dagger \sum_{j=0}^{q} l_{i,j} s^j = \sum_{j=0}^{q} \left(\sum_{i=0}^{n} A_i^\dagger l_{i,j} \right) s^j = \sum_{j=0}^{q} R_j^\dagger s^j \qquad (2.2.29)$$

gde je

$$R_j = \sum_{i=0}^{n} A_i^\dagger l_{i,j}, \qquad j = 0, 1, \ldots, q. \qquad (2.2.30)$$

Opisani algoritam treba da se modifikuje samo u *Koraku 4.* na sledeći način:

Korak 4. Moore-Penrose inverz $A(s)^\dagger$ date matrice $A(s)$ je

$$A(s)^\dagger = \sum_{j=0}^{q} R_j^\dagger s^j, \quad \text{gde je } R_j = \sum_{i=0}^{n} A_i^\dagger l_{i,j}, j = 0, 1, \ldots, q. \qquad (2.2.31)$$

Za ovaj algoritam pogodno je da interpolacioni polinom tražimo u malom broju početnih tačaka. Za implementaciju u proceduralnim programskim jezicima, na primer u DELPHI (metod će biti opisam u sledećem poglavlju), pogodno je uzeti dve tačke s_0 i s_1 za interpolaciju. Za dve tačke postoji jedinstvena prava koja ih sadrži. Stoga je:

$$\begin{aligned}
L_0(s) &= \frac{s - s_1}{s_0 - s_1} = \frac{1}{s_0 - s_1} s + \frac{s_1}{s_1 - s_0} = l_{0,1} s + l_{0,0} \qquad (2.2.32)\\
L_1(s) &= \frac{s - s_0}{s_1 - s_0} = \frac{1}{s_1 - s_0} s + \frac{s_0}{s_0 - s_1} = l_{1,1} s + l_{1,0}\\
R_0 &= A_0^\dagger l_{0,0} + A_1^\dagger l_{1,0} = A_0^\dagger \frac{s_1}{s_1 - s_0} + A_1^\dagger \frac{s_0}{s_0 - s_1} \qquad (2.2.33)\\
R_1 &= A_0^\dagger l_{0,1} + A_1^\dagger l_{1,1} = A_0^\dagger \frac{1}{s_0 - s_1} + A_1^\dagger \frac{1}{s_1 - s_0}
\end{aligned}$$

Lagrange-ova interpolacija opisana u *Koraku 4*, *Algoritma 2.10*, može se implementirati na sledeći način:

```
Lagr[s_List, var_List, k_]:=
    Module[{n=Length[s],p},
      p=Product[(var[[1]]-s[[i]])/(s[[k]]-s[[i]]),{i,1,k-1}]*
        Product[(var[[1]]-s[[i]])/(s[[k]]-s[[i]]),{i,k+1,n}];
    Return[Simplify[p]]
]
```

Konačno, Moore-Penrose inverz date matrice $A(s)$ nalazimo kao:

```
Interpol[A_List, s_List] :=
    Module[{n=Length[s],var=Variables[A]},
        Do[A1[i]=A/.var[[1]]->s[[i]],{i,1,n}];
        Do[A2[i]=Partitioning[A1[i]],{i,1,n}];
        A3=Sum[A2[i]*Lagr[s,var,i],{i,1,n}];
    Return[Expand[A3]//N]
]
```

Umesto opisane funkcije *Partitioning*[ ] može da se upotrebiti i ugrađena MATHEMATICA funkcija *PseudoInverse*[ ]. Ali korišćenje ove funkcije na matrice velikih dimenzija, zahteva mnogo više vremena za izračunavanje nego sa *Partitioning*[ ]. Takođe, u proceduralnim programskim jezicima,npr. DELPHI, ne postoji ugrađena funkcija za izračunavanje generalisanih inverza. Zato je Grevile rekurzivni algoritam neophodan.

Primer 2.36 Sledeći primer poznat je kao S_{11} iz [85]:

$$S11 = \begin{bmatrix} a+1 & a & a & a & a & a & a & a & a & a & a+1 \\ a & a-1 & a & a & a & a & a & a & a & a & a \\ a & a & a+1 & a & a & a & a & a & a & a & a \\ a & a & a & a-1 & a & a & a & a & a & a & a \\ a & a & a & a & a+1 & a & a & a & a & a & a \\ a & a & a & a & a & a-1 & a & a & a & a & a \\ a & a & a & a & a & a & a+1 & a & a & a & a \\ a & a & a & a & a & a & a & a-1 & a & a & a \\ a & a & a & a & a & a & a & a & a+1 & a & a \\ a & a & a & a & a & a & a & a & a & a-1 & a \\ a+1 & a & a & a & a & a & a & a & a & a & a+1 \end{bmatrix}.$$

Primenom Algoritam 2.10 i funkcije *Interpol*[ ] sa početnim tačkama za iteraciju $\{1, 2\}$ dobijamo:

In[**2**] := Interpol[S11, {1, 2}]

Out[2] = $\{\{0.25 - 0.25\,a, 0.5\,a, -0.5\,a, 0.5\,a, -0.5\,a, 0.5\,a, -0.5\,a, 0.5\,a, -0.5\,a, 0.5\,a, 0.25 - 0.25\,a\},$
$\{0.5\,a, -1. - 1.\,a, a, -1.\,a, a, -1.\,a, a, -1.\,a, a, -1.\,a, 0.5\,a\},$
$\{-0.5\,a, a, 1. - 1.\,a, a, -1.\,a, a, -1.\,a, a, -1.\,a, a, -0.5\,a\},$
$\{0.5\,a, -1.\,a, a, -1. - 1.\,a, a, -1.\,a, a, -1.\,a, a, -1.\,a, 0.5\,a\},$
$\{-0.5\,a, a, -1.\,a, a, 1. - 1.\,a, a, -1.\,a, a, -1.\,a, a, -0.5\,a\},$
$\{0.5\,a, -1.\,a, a, -1.\,a, a, -1. - 1.\,a, a, -1.\,a, a, -1.\,a, 0.5\,a\},$
$\{-0.5\,a, a, -1.\,a, a, -1.\,a, a, 1. - 1.\,a, a, -1.\,a, a, -0.5\,a\},$
$\{0.5\,a, -1.\,a, a, -1.\,a, a, -1.\,a, a, -1. - 1.\,a, a, -1.\,a, 0.5\,a\},$
$\{-0.5\,a, a, -1.\,a, a, -1.\,a, a, -1.\,a, a, 1. - 1.\,a, a, -0.5\,a\},$
$\{0.5\,a, -1.\,a, a, -1.\,a, a, -1.\,a, a, -1.\,a, a, -1. - 1.\,a, 0.5\,a\},$
$\{0.25 - 0.25\,a, 0.5\,a, -0.5\,a, 0.5\,a, -0.5\,a, 0.5\,a, -0.5\,a, 0.5\,a, -0.5\,a, 0.5\,a, 0.25 - 0.25\,a\}\}$

2.2.5 Drazinov inverz sa više promenljivih

Predpostavimo da je $A(s_1, s_2) \in \mathbb{C}(s_1, s_2)^{n\times n}$ realna matrica sa dve promenljive. Dokaz sledećeg tvrđenja je sličan odgovarajućem u [22, 23].

Lema 2.4 *Razmotrimo polinomijalnu matricu sa dve promenljive $A(s_1, s_2)$. Predpostavimo da je*

$$\begin{aligned} a(z, s_1, s_2) &= \det\left[zI_n - A(s_1, s_2)\right] = \sum_{l=0}^{n} a_l(s_1, s_2) z^{n-l}, \\ a_0(s_1, s_2) &\equiv 1, \quad z \in \mathbb{C} \end{aligned}$$

karakteristični polinom za $A(s_1, s_2)$. Takođe, uočimo sledeći niz realnih matrica dimenzije $n \times n$.

$$B_j(s_1, s_2) = \sum_{l=0}^{n} a_l(s_1, s_2) A(s_1, s_2)^{j-l}, \quad j = 0, \ldots, n.$$

Neka je

$$a_n(s_1, s_2) \equiv 0, \ldots, a_{t+1}(s_1, s_2) \equiv 0, \quad a_t(s_1, s_2) \neq 0.$$

Takođe, predpostavimo da je

$$B_n(s_1, s_2) \equiv \mathbb{O}, \ldots, B_r(s_1, s_2) \equiv \mathbb{O}, \quad B_{r-1}(s_1, s_2) \neq \mathbb{O}$$

i $k = r - t$.

U slučaju da je $k > 0$, Drazinov inverz za $A(s_1, s_2)$ dat je sa

$$\begin{aligned} A(s_1, s_2)^D &= a_t(s_1, s_2)^{-k-1} A(s_1, s_2)^k B_{t-1}(s_1, s_2)^{k+1} \\ &= a_t(s_1, s_2)^{-k-1} A(s_1, s_2)^k \left[\sum_{l=0}^{t-1} a_l(s_1, s_2) A(s_1, s_2)^{t-1-l}\right]^{k+1}. \end{aligned}$$

U slučaju da je $k = 0$ imamo $A(s_1, s_2)^D \equiv \mathbb{O}$.

Na osnovu rezultata *Leme 2.4* može se postaviti sledeća generalizacija *Algoritma 2.7*

Algoritam 2.11 *Drazinov inverz date realne matrice $A(s_1, s_2) \in \mathbb{C}(s_1, s_2)^{n\times n}$.*

Korak 1. Konstruisati niz brojeva $\{a_0(s_1, s_2), \ldots, a_n(s_1, s_2)\}$ i niz realnih matrica $\{B_0(s_1, s_2), \ldots, B_n(s_1, s_2)\}$ kao u sledećem:

$$
\begin{array}{ll}
A_0(s_1,s_2)=\mathbb{O}, & a_0(s_1,s_2)=1,\\
B_0(s_1,s_2)=I_n & \\
A_1(s_1,s_2)=A(s_1,s_2)B_0(s_1,s_2), & a_1(s_1,s_2)=-\frac{Tr(A_1(s_1,s_2))}{1},\\
B_1(s_1,s_2)=A_1(s_1,s_2)+a_1(s_1,s_2)I_n & \\
\dots & \dots\\
A_n(s_1,s_2)=A(s_1,s_2)B_{n-1}(s_1,s_2), & a_n(s_1,s_2)=-\frac{Tr(A_n(s_1,s_2))}{n},\\
B_n(s_1,s_2)=A_n(s_1,s_2)+a_n(s_1,s_2)I_n. &
\end{array}
\tag{2.2.34}
$$

Korak 2. Neka je

$$t=\max\{l : a_l(s_1,s_2)\neq 0\},\quad r=\min\{l : B_l(s_1,s_2)=\mathbb{O}\},\quad k=r-t.$$

Drazinov inverz $A(s_1,s_2)^D$ dat je sa

$$A(s_1,s_2)^D = a_t(s_1,s_2)^{-k-1}A(s_1,s_2)^k B_{t-1}(s_1,s_2)^{k+1}. \tag{2.2.35}$$

Predpostavimo da je $A[s_1,s_2]$ polinomijalna matrica u formi

$$A[s_1,s_2]=\sum_{j_1=0}^{q_1}\sum_{j_2=0}^{q_2} A_{j_1,j_2}s_1^{j_1}s_2^{j_2}\in\mathbb{C}[s_1,s_2]^{n\times n} \tag{2.2.36}$$

gde su $A_{j_1,j_2}\in\mathbb{C}^{n\times n}$, $j_1=0,\dots,q_1$, $j_2=0,\dots,q_2$ konstantne kompleksne matrice i s_1, s_2 nepoznate promenljive.

Koeficijenti $a_i(s_1,s_2)$ mogu se posmatrati kao polinomi sa promenljivima s_1 i s_2. Takođe, matrični koeficijenti $B_i(s_1,s_2)$ mogu se smatrati polinomijalnim matricama uz stepene za s_1 i s_2. Ovaj pristup daje analognu reprezentaciju Drazinovog inverza i odgovarajući algoritam za izračunavanje Drazinovog inverza.

Teorema 2.24 *Neka je $A[s_1,s_2]$ polinomijalna matrica u formi* (2.2.34). *Neka su konstante $\{a_0(s_1,s_2),\dots,a_n(s_1,s_2)\}$ i $\{B_0(s_1,s_2),\dots,B_n(s_1,s_2)\}$, niz polinomijalnih matrica sa dve promenljive koje su definisane u Algoritam 2.8. Napišimo $a_i(s_1,s_2)$ i $B_i(s_1,s_2)$ kao*

$$a_i(s_1,s_2) = \sum_{j_1=0}^{iq_1}\sum_{j_2=0}^{iq_2} a_{i,j_1,j_2}s_1^{j_1}s_2^{j_2},\quad i=0,\dots,n, \tag{2.2.37}$$

$$B_i(s_1,s_2) = \sum_{j_1=0}^{iq_1}\sum_{j_2=0}^{iq_2} B_{i,j_1,j_2}s_1^{j_1}s_2^{j_2},\quad i=0,\dots,n, \tag{2.2.38}$$

gde su a_{i,j_1,j_2} skalari i B_{i,j_1,j_2} matrice sa konstantnim koeficijentima uz stepene $s_1^{j_1}$ i $s_2^{j_2}$, za svako $i = 0, \dots, n$, $j_1 = 0, \dots, iq_1$, $j_2 = 0, \dots, iq_2$. Onda su skalari a_{i,j_1,j_2} i matrice B_{i,j_1,j_2} generisane sa:

$$
\begin{aligned}
a_{0,j_1,j_2} &= 1,\\
a_{i+1,j_1,j_2} &= -\frac{1}{i+1}\mathrm{Tr}\left(\sum_{l_1=0}^{j_1}\sum_{l_2=0}^{j_2} A_{j_1-l_1,j_2-l_2}B_{i,l_1,l_2}\right), \qquad (2.2.39)\\
i &= 0,\dots,n-1, \quad j_1 = 0,\dots,(i+1)q_1, \ j_2 = 0,\dots,(i+1)q_2.\\
B_{0,j_1,j_2} &= I_n,\\
B_{i+1,j_1,j_2} &= \sum_{l_1=0}^{j_1}\sum_{l_2=0}^{j_2} A_{j_1-l_1,j_2-l_2}B_{i,l_1,l_2} + a_{i+1,j_1,j_2}I_n, \qquad (2.2.40)\\
i &= 0,\dots,n-1, \quad j_1 = 0,\dots,(i+1)q_1, \ j_2 = 0,\dots,(i+1)q_2.
\end{aligned}
$$

Dokaz. Nije teško pokazati iz (2.2.34) da su najveći stepeni za $A_i(s_1, s_2)$ (a time i za $B_i(s_1, s_2)$) saglasno promenljivima s_1 i s_2 jednaki iq_1 i iq_2, respektivno, za svako $i = 0, \dots, n-1$. Takođe, stepeni polinomijalnih veličina $a_i(s_1, s_2)$, $i = 1, \dots, n$ su jednaki iq_1 i iq_2. Otuda se $a_i(s_1, s_2)$ i $B_i(s_1, s_2)$ mogu napisati u formi (2.2.37) i (2.2.38), respektivno. Takođe, iz (2.2.34) i (2.2.36), dobija se

$$
\begin{aligned}
A_{i+1}(s_1, s_2) &= A(s_1, s_2)B_i(s_1, s_2)\\
&= \left(\sum_{j_1=0}^{q_1}\sum_{j_2=0}^{q_2} A_{j_1,j_2}s_1^{j_1}s_2^{j_2}\right)\left(\sum_{l_1=0}^{iq_1}\sum_{l_2=0}^{iq_2} B_{i,l_1,l_2}s_1^{l_1}s_2^{l_2}\right) \qquad (2.2.41)\\
&= \sum_{j_1=0}^{(i+1)q_1}\sum_{j_2=0}^{(i+1)q_2}\left(\sum_{l_1=0}^{j_1}\sum_{l_2=0}^{j_2} A_{j_1-l_1,j_2-l_2}B_{i,l_1,l_2}\right)s_1^{j_1}s_2^{j_2},
\end{aligned}
$$

za svako $i = 0, \dots, n-1$. Primenom (2.2.34) i (2.2.41) sledi da je

$$
\begin{aligned}
a_{i+1}(s_1, s_2) &= -\frac{1}{i+1}\mathrm{Tr}(A_{i+1}(s_1, s_2)) \qquad (2.2.42)\\
&= \sum_{j_1=0}^{(i+1)q_1}\sum_{j_2=0}^{(i+1)q_2}\left[-\frac{1}{i+1}\mathrm{Tr}\left(\sum_{l_1=0}^{j_1}\sum_{l_2=0}^{j_2} A_{j_1-l_1,j_2-l_2}B_{i,l_1,l_2}\right)\right]s_1^{j_1}s_2^{j_2}.
\end{aligned}
$$

Jednakost (2.2.39) sledi iz (2.2.37) i (2.2.42).

Sa druge strane, koristeći (2.2.34), (2.2.37) i (2.2.41), za svako $i = 0, \dots, n-1$ dobija se

$$
\begin{aligned}
B_{i+1}(s_1, s_2) &= A_{i+1}(s_1, s_2) + a_{i+1}(s_1, s_2)I_n\\
&= \sum_{j_1=0}^{(i+1)q_1}\sum_{j_2=0}^{(i+1)q_2}\left(\sum_{l_1=0}^{j_1}\sum_{l_2=0}^{j_2} A_{j_1-l_1,j_2-l_2}B_{i,l_1,l_2}\right)s_1^{j_1}s_2^{j_2} + \sum_{j_1=0}^{(i+1)q_1}\sum_{j_2=0}^{(i+1)q_2} a_{i+1,j_1,j_2}I_n s_1^{j_1}s_2^{j_2}\\
&= \sum_{j_1=0}^{(i+1)q_1}\sum_{j_2=0}^{(i+1)q_2}\left(\sum_{l_1=0}^{j_1}\sum_{l_2=0}^{j_2} A_{j_1-l_1,j_2-l_2}B_{i,l_1,l_2} + a_{i+1,j_1,j_2}I_n\right)s_1^{j_1}s_2^{j_2} \qquad (2.2.43)
\end{aligned}
$$

Jednakost (2.2.40) sledi iz (2.2.38) i (2.2.43). □

Sada smo u mogućnosti da postavimo sledeći algoritam za izračunavanje Drazinovog inverza $A(s_1,s_2)^D$. Ovaj algoritam može se smatrati kao prirodni nastavak algoritma za izračunavanje Moore-Penrose inverza, opisanog u [22, 23] i generalizaciji Grevileovog algoritma iz [30, 31].

Algoritam 2.12 Drazinov inverz za $A\in\mathbb{C}[s_1,s_2]^{n\times n}$.
Inicijalni uslovi:

$$B_{0,0,0}=I_n, \quad A_{j_1,j_2}=0, \quad j_1=q_1+1,\dots,nq_1,\ j_2=q_2+1,\dots,nq_2.$$

Granični uslovi:

$$\begin{array}{rcl} B_{0,j_1,j_2} & = & \mathbb{O} \quad \forall j_1\in\mathbb{N},\ j_2\in\mathbb{N} \\ B_{i,j_1,j_2} & = & \mathbb{O},\quad i=0,\dots,n-1,\ j_1=iq_1+1,\dots,(n-1)q_1,\ j_1=iq_1+1,\dots,(n-1)q_1, \\ a_{0,j_1,j_2} & = & 1, j_1=0,\dots,q_1,\ j_2=0,\dots,q_2. \end{array}$$

Rekurzivni koraci za $a_i(s_1,s_2)$, $1\le i\le n-1$ dati su u (2.2.39).

Rekurzivne relacije za B_{ij_1,j_2}, $1\le i\le n-1$ date su u (2.2.40).

Kriterijum zaustavljanja: Izračunati prvi t koji zadovoljava

$$a_{t+1,j_1,j_2}=a_{t+2,j_1,j_2}=\cdots=a_{n,j_1,j_2}=0, \quad \forall j_1,j_2\in\mathbb{N}.$$

i prvi r koji zadovoljava

$$B_{r,j_1,j_2}=\mathbb{O}, \quad j_1=0,\dots,rq_1,\ j_2=0,\dots,rq_2.$$

Za $k=r-t$ postaviti

$$\begin{array}{rcl} B_{j_1,j_2} & = & B_{t-1,j_1,j_2}, \quad j_1=0,\dots,(t-1)q_1,\ j_2=0,\dots,(t-1)q_2, \\ a_{j_1,j_2} & = & \hat{a}_{t,j_1,j_2}, \quad j_1=0,\dots,tq_1,\ j_2=0,\dots,tq_2. \end{array}$$

IZLAZ:

$$\begin{aligned} A(s_1,s_2)^D = & \left(\sum_{j_1=0}^{tq_1}\sum_{j_2=0}^{tq_2} a_{j_1,j_2}s_1^{j_1}s_2^{j_2}\right)^{-k-1}\left(\sum_{i_1=0}^{q_1}\sum_{i_2=0}^{q_2} A_{i_1,i_2}s_1^{i_1}s_2^{i_2}\right)^{k}\times \\ & \times\left(\sum_{l_1=0}^{(t-1)q_1}\sum_{l_2=0}^{(t-1)q_2} B_{l_1,l_2}s_1^{l_1}s_2^{l_2}\right)^{k+1}. \end{aligned}$$

Opišimo sad implementaciju *Algoritma 2.12.* Matrica $A[s_1, s_2]$ može se predstaviti kao tro-dimenzionalna lista $\{A_1, \ldots, A_{(q_1+1)(q_2+1)}\}$ matrica. Ograničimo implementaciju na skup polinomijalnih matrica sa realnim koeficijentima da bi obezbedili pojednostavljenje pomoću standardne MATHEMATICA funkcije *Simplify*. Nekoliko pomoćnih procedura opisane su na početku.

Funkcija $FrmA2[A]$ transformiše matricu $A[s_1, s_2]$ u $\{A_1, \ldots, A_{(q_1+1)(q_2+1)}\}$.

```
FrmA2[a_List]:=
   Block[{L={},L1,nul,var1,var2,n1,n2,m1,i,j,k,koef,Lp},
     {n1,m1}=Dimensions[a];
     nul=Table[0,{i,n1},{j,m1}];
     L1=nul;
     var=Variables[a];var1=var[[1]];var2=var[[2]];
     n1=Max[Exponent[a,var1,List]];  n2=Max[Exponent[a,var2,List]];
     For[i=0,i<=n1,i++,
       For[j=0,j<=n2,j++,
          If[(i=!=0)||(j=!=0),
            koef=Coefficient[a,var1^i*var2^j];
            If[Variables[koef]=!={},
               Lp=Position[koef,Variables[koef][[1]]];
               For[k=1,k<=Length[Lp],k++,
                   koef[[Lp[[k,1]],Lp[[k,2]]]]=0
            ]; ];
            AppendTo[L,koef];    L1=L1+koef*var1^i*var2^j;
     ] ] ];
     L=Prepend[L,a-L1]; Return[L]
   ]
```

Iz interne reprezentacije $\{A_1, \ldots, A_{(q_1+1)(q_2+1)}\}$ moguće je naći konstantne matrice A_{j_1,j_2} koje odgovaraju stepenima uz $s_1^{j_1} s_2^{j_2}$ pomoću funkcije

$$A2[j_1, j_2] = \begin{cases} A[[j_1(q_2+1) + j_2 + 1]], & j_1 \le q_1, j_2 \le q_2, \\ 0, & \text{u suprotnom} \end{cases}$$

koja se implementira kao

```
A2[j1_,j2_]:=If[(j1<=q1)&&(j2<=q2),Return[A[[j1(q2+1)+j2+1]]],
                                    Return[Table[0,{n},{n}]]]
```

Rezultat pomoćne funkcije `Tr[A]` je trag kvadratne matrice A.

```
Tr[a_List]:=
    Module[{ i,n1,m1},
    {n1,m1}=Dimensions[a];
    Return[Sum[a[[i,i]], {i,n1}]]
```

```
]
```

Rekurzivne relacije (2.2.39) i (2.2.40) za izračunavanje realnih brojeva $a_{i,j1,j2}$ i matrica $B_{i,j1,j2}$, respektivno, implementirana su sledećim dvema funkcijama.

```
Fa2[i1_,j1_,j2_]:=-1/i1*Tr[Sum[Sum[A2[j1-l1,j2-l2]
                        .B2[i1-1,l1,l2],{l2,0,j2}],{l1,0,j1}]]
B2[u_,v1_,v2_]:=B2[u,v1,v2]=Block[{w={},j1,j2,nul},
   nul=Table[0,{n},{n}];
   If[u==0 && v1==0 && v2==0,w=IdentityMatrix[n],
     If[u==0,w=nul,
      If[((v1>=u q1+1)&&(v1<=(n-1) q1))&&((v2>=u q2+1)
                      &&(v2<=(n-1) q2)),w=nul,
       w=Sum[Sum[A2[v1-j1,v2-j2].B2[u-1,j1,j2],{j2,0,v2}],
                {j1,0,v1}]+ Fa2[u,v1,v2]*IdentityMatrix[n]
   ]]];
   Return[w]
]
```

Algoritam 2.12 je implementiran u funkciji *DrzPoly2[mat_List]*, gde se predpostavlja da formalni parametar *mat_* predstavlja $\{A_1, \dots, A_{(q1+1)(q2+1)}\}$ listu koja odgovara polinomijalnoj matrici.

```
DrzPoly2[mat_List]:=
   Block[{A,i,j,j1,j2,s,k,r=t=0,log1=log2=True,var,var1,var2,q1,q2},
   var=Variables[mat];   var1=var[[1]];   var2=var[[2]];
   If[var =={}, A=mat,A=FrmA2[mat]];
   q1=Max[Exponent[mat,var1,List]]; q2=Max[Exponent[mat,var2,List]];
   {n,n}=Dimensions[A[[1]]]; nul=Table[0,{n},{n}];
   For[i=0,i<=n-1,i++,
     For[j1=0,j1<=(i+1) q1,j1++,
       For[j2=0,j2<=(i+1) q2,j2++,
         If[Fa2[i+1,j1,j2]===0 ,log1=log1 && True,log1=log1 && False];
         If[B2[i+1,j1,j2]===nul ,log2=log2 && True,log2=log2 && False];
     ] ];
     If[log1,t=i,log1=True]; If[log2,r=i+1,log2=True]];
   If[t==0, Return[Simplify[Inverse[mat]]]]; (* Regular matrix *)
   k=r-t; (* Singular matrix *)
   rez=Sum[Sum[Fa2[r-1,j1,j2]*var1^j1*var2^j2,{j2,0,t q2}],{j1,0,t q1}]^(-k-1)*
    MatrixPower[Sum[Sum[A2[j1,j2]*var1^j1*var2^j2,{j2,0,q2}],{j1,0,q1}],k].
    MatrixPower[Sum[Sum[B2[t-1,l1,l2]*var1^l1*var2^l2,
                         {l2,0,(t) q2}],{l1,0,(t-1) q1}],k+1];
   Return[MatrixForm[Simplify[rez]]]
```

Primer 2.37 Interna reprezentacija polinomijalne matrice sa dve promenljive $X = \begin{bmatrix} 2w & 1 & w-s \\ s^2 & 3w^3 & ws^2 \\ 2 & s^2-3w & s \end{bmatrix}$, može biti generisana izrazom

In[1]:=FrmA2[X]

`Out[1]=` {{{0,1,0},{0,0,0},{2,0,0}},{{2,0,1},{0,0,0},{0,-3,0}},{{0,0,0},{0,0,0},{0,0,0}}, {{0,0,0},{0,3,0},{0,0,0}},{{0,0,-1},{0,0,0},{0,0,1}},{{0,0,0},{0,0,0},{0,0,0}}, {{0,0,0},{0,0,0},{0,0,0}},{{0,0,0},{0,0,0},{0,0,0}},{{0,0,0},{1,0,0},{0,1,0}},

{{0,0,0},{0,0,1},{0,0,0}},{{0,0,0},{0,0,0},{0,0,0}},{{0,0,0},{0,0,0},{0,0,0}}}

Primer 2.38 Za polinomijalnu matricu sa dve promenljive $X = \begin{bmatrix} w+1 & s & w+1 \\ w1 & w-1 & w \\ w+1 & w & w+1 \end{bmatrix}$ funkcija $DrzPoly2[\,]$ daje Drazinov inverz:

In[2]:=DrzPoly2[X]

Out[2]=

$$\begin{bmatrix} \frac{1-w+2\,w^3-2\,w^2\,w1}{\left(2-w^2+w\,w1\right)^2} & \frac{-3\,w^3+2\,w^2\,(-2+w1)+3\,w1+w\,\left(-1+4\,w1+w1^2\right)}{\left(2-w^2+w\,w1\right)^2} & \frac{1-w+2\,w^3-2\,w^2\,w1}{\left(2-w^2+w\,w1\right)^2} \\ \frac{w}{2-w^2+w\,w1} & \frac{2\,(1+w)}{-2+w^2-w\,w1} & \frac{w}{2-w^2+w\,w1} \\ \frac{1-w^3+w\,(-1+w1)+w^2\,(-1+w1)}{\left(2-w^2+w\,w1\right)^2} & \frac{2\,w^3+w\,(3-4\,w1)-2\,w^2\,(-2+w1)-w1}{\left(2-w^2+w\,w1\right)^2} & \frac{1-w^3+w\,(-1+w1)+w^2\,(-1+w1)}{\left(2-w^2+w\,w1\right)^2} \end{bmatrix}$$

Primer 2.39 Razmotrimo matricu sa dve promenljive w i s:

$$X = \begin{bmatrix} 1 & w & 1 & w & 1 \\ 1 & s & 1 & s & 1 \\ s & w & s^2 & w & s \\ s & w^2 & s & w^2 & s \\ s^3 & w & s & w & s \end{bmatrix}$$

In[3]:=DrzPoly2[X]

Out[3]=
$$\left[\begin{array}{ccc} \frac{s^2-w+s(-1+w)w}{(-1+s)s(1+s)(s-w)(-1+w)} & -\frac{w}{s(1+s)(s-w)(-1+w)} & 0 \\ \frac{-s^4-w^2+s(-1+w)^2(1+w)+s^2(1+w^2)}{(-1+s)s(1+s)(s-w)(-1+w)^2} & \frac{s+s^2-sw-w^2}{s(1+s)(s-w)(-1+w)^2} & 0 \\ \frac{s^2-w+s(-1+w)w}{(-1+s)s(s-w)(-1+w)} & -\frac{w}{s(s-w)(-1+w)} & \frac{1}{-s+s^2} \\ \frac{s^3(-1+w)+s^4w+w^2-s(-1+w)w^2-s^2w(1+w)}{(-1+s)s(1+s)(s-w)(-1+w)^2} & & \\ -\frac{s^4-2w+s(-3+w)w+s^3(1+w^2)+s^2(1-w+w^2)}{(-1+s)s(1+s)(s-w)(-1+w)} & \frac{(2+2s+s^2)w}{s(1+s)(s-w)(-1+w)} & \frac{1}{s-s^2} \end{array} \right.$$
$$\left. \begin{array}{cc} -\frac{w}{s(1+s)(s-w)(-1+w)} & \frac{1}{-s+s^3} \\ \frac{s+s^2-sw-w^2}{s(1+s)(s-w)(-1+w)^2} & \frac{s+w}{(-s+s^3)(-1+w)} \\ -\frac{w}{s(s-w)(-1+w)} & 0 \\ \frac{w(-s^2+w)}{s(1+s)(s-w)(-1+w)^2} & -\frac{s+w}{(-s+s^3)(-1+w)} \\ \frac{w(-s^2+w)}{s(1+s)(s-w)(-1+w)^2} & 0 \\ \frac{(2+2s+s^2)w}{s(1+s)(s-w)(-1+w)} & \frac{1}{s-s^3} \end{array} \right]$$

2.2.6 Konačni algoritam za polinomijalne matrice

Implementacija *Algoritma 2.4* u terminima simboličke manipulacije dostupnih funkcija iz jezika MATHEMATICA proizvodi dobro poznat poteškoće simboličke implementacije opisane u [79], kao što su sporo izvršavanje i veliki memorijski prostor potreban za numeričku implementaciju. Zbog toga, u ovom delu zrazvijen je numerički algoritam koji odgovara *Algoritmu 2.4* u slučaju kad je $A(s)$ polinomijalna matrica. Na ovaj način se generalizuje poznati algoritam za izračunavanje Moore-Penrose inverza iz [23] i analogni algoritam za izračunavanje Drazin inverza iz [20] i [75].

Predpostavimo da su $R(s) \in \mathbb{C}[s]^{m\times n}$ and $R(s) \in \mathbb{C}[s]^{m\times n}$ polinomijalne matrice u formi

$$\begin{aligned} R(s) &= R_r s^r + R_{r-1}s^{r-1} + \cdots + R_1 s + R_0, \\ T(s) &= T_t s^t + T_{t-1}s^{t-1} + \cdots + T_1 s + T_0, \end{aligned} \tag{2.2.44}$$

gde su $R_i \in \mathbb{C}^{m\times n}$, $i = 0, \ldots, r$ i $T_i \in \mathbb{C}^{m\times n}$, $i = 0, \ldots, t$ kostantnet kompleksne matrice dimenzije $m \times n$, i s nepoznata promenljiva. Koeficijenti $a_i(s)$ mogu biti posmatrani kao polinomi od s. Takođe, koeficijenti matrice $B_i(s)$ su polinomijalne matrice saglasno stepenu uz promenljivu s. Ovaj pristup daje nam analognu reprezentaciju matrice $X_e(s)$ definisane u (2.1.7), i analognog algoritma za njegovo izračunavanje. Koristeći relacije (2.1.1), (2.1.2) i (2.1.3), lako se pokazuje indukcijom da su stepeni s u $a_i(s)$ i $B_i(s)$ skoro jednaki $i * (r + t)$. Zato se, skalarni polinomi $a_i(s)$ i polinomijalne matrice $B_i(s)$ mogu napisati na sledeći način:

$$a_i(s) = \sum_{j=0}^{i(r+t)} a_{i,j} s^j, \quad i = 1, \ldots, m, \tag{2.2.45}$$

$$B_i(s) = \sum_{j=0}^{i(r+t)} B_{i,j} s^j, \quad i = 1, \ldots, m, \tag{2.2.46}$$

gde su $a_{i,j}$, $i = 1, \ldots, m$, $j = 0, \ldots, i(r+t)$ skalari a $B_{i,j}$, $i = 1, \ldots, m$, $j = 0, \ldots, i(r+t)$ konstantne matrice dimenzije $m \times n$, koje odgovaraju stepenima s^j.

Teorema 2.25 *Neka su matrice* $R(s)$ *i* $T(s)$ *definisane sa* (2.2.44). *Posmatrajmo polinome* $\{a_0(s), \ldots, a_m(s)\}$ *i* $\{B_0(s), \ldots, B_m(s)\}$ *niz polinomijalnih matrica, definisani kao u Algoritmu 2.4. Neka je* $a_i(s)$ *i* $B_i(s)$ *kao u* (2.2.45) *i* (2.2.46), *respektivno. Onda je*

$$
\begin{aligned}
a_{i+1,j} &= -\frac{1}{i+1}\mathrm{Tr}\left(\sum_{k=0}^{j}\left(\sum_{l=0}^{j-k} T_{j-k-l}R_l^*\right)B_{i,k}\right),\\
&\quad i=0,1,\dots,m-1, \quad j=0,1,\dots,(i+1)*(r+t), \qquad (2.2.47)\\
B_{i+1,j} &= \sum_{k=0}^{j}\left(\sum_{l=0}^{j-k} T_{j-k-l}R_l^*\right)B_{i,k}+a_{i+1,j}I_p,\\
&\quad i=0,1,\dots,m-1, \quad j=0,1,\dots,(i+1)*(r+t), \qquad (2.2.48)
\end{aligned}
$$

gde se u (2.2.47) *i* (2.2.48) *predpostavlja da je* $R_u=\mathbb{O}$, $u\ge r+1$, $T_u=\mathbb{O}$, $u\ge t+1$ *i* $B_{iu}=\mathbb{O}$, $u\ge i*(t+r)+1$.

Dokaz. Iz (2.1.2), (2.2.44) i (2.2.46), dobijamo sledeće

$$
\begin{aligned}
A_{i+1}(s) &= T(s)R^*(s)B_i(s)=\left(\sum_{j=0}^{t}T_js^j\right)\left(\sum_{k=0}^{r}R_k^*s^k\right)\left(\sum_{l=0}^{i(r+t)}B_{i,l}s^l\right)\\
&= \left(\sum_{j=0}^{r+t}\left(\sum_{k=0}^{j}T_{j-k}R_k^*\right)s^j\right)\left(\sum_{l=0}^{i(r+t)}B_{i,l}s^l\right) \qquad (2.2.49)\\
&= \sum_{j=0}^{(i+1)(r+t)}\left(\sum_{k=0}^{j}\left(\sum_{l=0}^{j-k}T_{j-k-l}R_l^*\right)B_{i,k}\right)s^j.
\end{aligned}
$$

Zamenom (2.2.49) u (2.1.3) i koristeći (2.2.45) dobijamo (2.2.47).

Sa druge strane, koristeći (2.1.3), (2.2.45) i (2.2.49), dobijamo

$$
\begin{aligned}
B_{i+1}(s) &= A_{i+1}(s)+a_{i+1}(s)I_n \qquad (2.2.50)\\
&= \sum_{j=0}^{(i+1)(r+t)}\left(\sum_{k=0}^{j}\left(\sum_{l=0}^{j-k}A_{j-k-l}R_l^*\right)B_{i,k}\right)s^j+\sum_{j=0}^{(i+1)(r+t)}a_{i+1,j}I_ns^j\\
&= \sum_{j=0}^{(i+1)(r+t)}\left(\sum_{k=0}^{j}\left(\sum_{l=0}^{j-k}A_{j-k-l}R_l^*\right)B_{i,k}+a_{i+1,j}I_n\right)s^j.
\end{aligned}
$$

Jednakost (2.2.48) sledi iz (2.2.46) i (2.2.50). □

Sledi algoritam za izračunavanje polinomijane matrice $X_e(s)$. Ovaj algoritam je numerički ekvivalentan Algoritmu 2.4, i primenljiv na polinomijalne matrice.

Algoritam 2.13 *Data je matrica* $A(s)\in\mathbb{C}[s]^{m\times n}$. *Izabrati* $R(s),T(s)\in\mathbb{C}[s]^{m\times n}$ *u formi* (2.2.44).

Početni uslovi:

$$
\begin{aligned}
B_{0,0} &= I_n,\quad R_u=0,\ u=r+1,\dots,m*(r+t),\ T_u=0,\ u=t+1,\dots,m*(r+t)\\
a_{0,0} &= 1,\ a_{0,j}=0,\ j=1,\dots,m(r+t).
\end{aligned}
$$

Ograničenja:

$$\begin{aligned} B_{0,j} &= \mathbb{O} \quad \forall j \geq 1, \\ B_{i,j} &= \mathbb{O}, \quad i{=}0,\ldots,m-1,\ j{=}i*(r+t)+1,\ldots,(m-1)*(r+t). \end{aligned}$$

Glavni algoritam:

$i = 0$

DO WHILE najmanje jednom $B_{i,j} <> 0,\ j = 0,1,\ldots,(i+1)*(r+t)$*:*

Početni uslovi za B_{ij}*:*

$B_{i,j} = \mathbb{O},\ j = i*(r+t)+1,\ldots,(n-1)*(r+t).$

Rekurzivne relacije za $a_i(s)$ *su definisane u* (2.2.47).

Rekurzivne relacije za $B_i(s)$ *su definisane u* (2.2.48).

$i = i + 1.$

END DO.

Uslov prekida: Izračunati prvo k *koje zadovoljava*

$$a_{k+1,j} = a_{k+2,j} = \cdots = a_{r+t,j} = 0 \quad \forall j \in \mathbb{N}.$$

Izlaz: Vraća polinomijalnu matricu

$$X_e(s){=}(-1)^e * \left(\sum_{j=0}^{k*(r+t)} a_{k,j}s^j\right)^{-e} \left(\sum_{i=0}^{r} R_i^* s^i\right) \left(\sum_{l=0}^{(k-1)*(r+t)} B_{k-1,l}s^l\right)^{e}. \tag{2.2.51}$$

Evidentno je da samo željena simbolička manipulacija u *Algoritmu 2.13*, postignuta simplifikacijom elemenata rezultujuće matrice $X_e(s)$.

Primer 2.40 Neka je data polinomijalna matrica $A(s)$ kao u (2.1.10). Da bi generisali traženi inverz za $A(s)$ izaberimo matrice $R(s)$ i $T(s)$ kao:

$$R(s) = \begin{bmatrix} 1+s+2s^2 & 3s & 1-s+2s^2 \\ 0 & 0 & 0 \\ 1+s+2s^2 & 3s & 1-s+2s^2 \end{bmatrix}, \quad T(s) = A(s).$$

Konstantne matrice R_i i T_i su

$$R_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad R_1 = \begin{bmatrix} 1 & 3 & -1 \\ 0 & 0 & 0 \\ 1 & 3 & -1 \end{bmatrix} R_2 = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 2 \end{bmatrix}.$$

Kako je $r + t = 4$, početni uslovi su:

$$\begin{aligned} &R_3 = R_4 = \mathbb{O}, \quad T_3 = T_4 = \mathbb{O}, \\ &a_{0,0} = 1,\ a_{0,1} = a_{0,2} = a_{0,3} = a_{0,4} = a_{0,5} = a_{0,6} = 0, \\ &B_{0,0} = I_3,\ B_{0,1} = B_{0,2} = B_{0,3} = B_{0,4} = B_{0,5} = B_{0,6} = \mathbb{O}. \end{aligned}$$

U slučaju $i = 0$, gde je sa $T_i.R_j^T$ označen proizvod matrica T_i i R_j^T, a Tr označen trag date matrice, imamo sledeće:

$$
\begin{aligned}
a_{1,0} &= -1/(i+1) * \mathrm{Tr}[T_0.R_0^T.B_{0,0}] = 0,\\
a_{1,1} &= -1/(i+1) * \mathrm{Tr}[(T_1.R_0^T + T_0.R_1^T).B_{0,0} + T_0.R_0^T.B_{0,1}] = -10,\\
a_{1,2} &= -1/(i+1) * \mathrm{Tr}[(T2.R_0^T + T_1.R_1^T + T_0.R_2^T).B_{0,0}+\\
&\quad (T_1.R_0^T + T_0.R_1^T).B_{0,1} + T_0.R_0^T.B_{0,2}] = -4,\\
a_{1,3} &= -1/(i+1) * \mathrm{Tr}[(T_3.R_0^T + T_2.R_1^T + T_1.R_2^T + T_0.R_3^T).B_{0,0}+\\
&\quad (T_2.R_0^T + T_1.R_1^T + T_0.R_2^T).B_{0,1}+\\
&\quad (T_1.R_0^T + T_0.R_1^T).B_{0,2} + T_0.R_0^T.B_{0,3}] = -8,\\
a_{1,4} &= -1/(i+1) * \mathrm{Tr}[(T_4.R_0^T + T_3.R_1^T + T_2.R_2^T + T_1.R_3^T + T_0.R_4^T).B_{0,0}+\\
&\quad (T_3.R_0^T + T_2.R_1^T + T_1.R_2^T + T_0.R_3^T).B_{0,1} + (T_2.R_0^T + T_1.R_1^T + T_0.R_2^T).B_{0,2}+\\
&\quad (T_1.R_0^T + T_0.R_1^T).B_{0,3} + T_0.R_0^T.B_{0,4}] = -8,
\end{aligned}
$$

$$
\begin{aligned}
B_{1,0} =& T_0.R_0^T + a_{1,0} * I_3 = \mathbb{O},\\
B_{1,1} =& (T_1.R_0^T + T_0.R_1^T).B_{0,0} + T_0.R_0^T.B_{0,1} + a_{1,1} * I_3 = \begin{bmatrix} -5 & 0 & 5\\ 0 & -10 & 0\\ 5 & 0 & -5 \end{bmatrix},\\
B_{1,2} =& (T_2.R_0^T + T_1.R_1^T + T_0.R_2^T).B_{0,0} + (T_1.R_0^T + T_0.R_1^T).B_{0,1} + T_0.R_0^T.B_{0,2} + a_{1,2} * I_3 =\\
& \begin{bmatrix} -2 & 0 & 2\\ 1 & -4 & 1\\ 2 & 0 & -2 \end{bmatrix},\\
B_{1,3} =& (T_3.R_0^T + T_2.R_1^T + T_1.R_2^T + T_0.R_3^T).B_{0,0} + (T_2.R_0^T + T_1.R_1^T + T_0.R_2^T).B_{0,1}+\\
& (T_1.R_0^T + T_0.R_1^T).B_{0,2} + T_0.R_0^T.B_{0,3} + a_{1,3} * I_3\\
=& \begin{bmatrix} -4 & 0 & 4\\ 3 & -8 & 3\\ 4 & 0 & -4 \end{bmatrix}\\
B_{1,4} =& (T_4.R_0^T + T_3.R_1^T + T_2.R_2^T + T_1.R_3^T + T_0.R_4^T).B_{0,0}+\\
& (T_3.R_0^T + T_2.R_1^T + T_1.R_2^T + T_0.R_3^T).B_{0,1}+\\
& (T_2.R_0^T + T_1.R_1^T + T_0.R_2^T).B_{0,2} + (T_1.R_0^T + T_0.R_1^T).B_{0,3} + T_0.R_0^T.B_{0,4} + a_{1,4} * I_3\\
=& \begin{bmatrix} -4 & 0 & 4\\ 0 & -8 & 0\\ 4 & 0 & -4 \end{bmatrix}.
\end{aligned}
$$

Za $i = 1$ imamo $a_{2,0} = a_{2,1} = \cdots = a_{2,8} = 0$, tj. $k = 1$. U skladu sa (2.2.51) za slučaj $e = 1$ dobijamo traženi inverz za $A(s)$:

$$
\begin{aligned}
X_e(s) &= -\left(a_{1,0} + a_{1,1}s + a_{1,2}s^2 + a_{1,3}s^3 + a_{1,4}s^4\right)^{-1} \left(\sum_{i=0}^{r} R_i^* s^i\right) B_{0,0}\\
&= (10s + 4s^2 + 8s^3 + 8s^4)^{-1} R(s) = \begin{bmatrix} \frac{1+s+2s^2}{10s+4s^2+8s^3+8s^4} & 0 & \frac{1+s+2s^2}{10s+4s^2+8s^3+8s^4}\\ \frac{3}{10+4s+8s^2+8s^3} & 0 & \frac{3}{10+4s+8s^2+8s^3}\\ \frac{1-s+2s^2}{10s+4s^2+8s^3+8s^4} & 0 & \frac{1-s+2s^2}{10s+4s^2+8s^3+8s^4} \end{bmatrix}.
\end{aligned}
$$

2.3 INVERZI RETKO POSEDNUTIH MATRICA

Operacije sa matricama, koje postoje u MATHEMATICA ne mogu se koristiti za rad sa retko posednutim matricama predstavljenim u vidu liste uređenih trojki $\{i, j, k\}$, gde je sa i označena vrsta, sa j kolona a sa k element date matrice na poziciji $\{i, j\}$ koji je različit od nule. Zbog te činjenice mora se napraviti kompletno nova algebra za rad sa ovakvim listama uređenih trojki, odnosno retko posednutim matricama.

Implementacija potrebnih funkcija.

$Nab[X]$ - predstavlja matricu X u obliku liste uređenih trojki (reprezentacija za retko posednute matrice);

```
Nab[a_List]:=
   Block[{a1=b1=l={}},
    {m,n}=Dimensions[a];
    For[i=1,i<=n,i++,
     For[j=1,j<=m,j++,
      If[a[[i,j]]=!=0,l=Append[l,{i,j,a[[i,j]]}]]]];
    Print["l=",l];
    For[i=1,i<=Length[l],i++,
     a1=Union[a1,{l[[i,1]]}]; b1=Union[b1,{l[[i,2]]}]];
    {a1,b1,l}];
```

$UzmiK[X, i]$ - iz matrice X uzima i-tu kolonu;

```
UzmiK[a_List,broj_]:=
   Block[{A=a,Ak={}},
    For[i=1,i<=Length[A],i++,
     If[A[[i,2]]==broj,Ak=Append[Ak,{A[[i,1]],1,A[[i,3]]}]]];
    Ak];
```

$MaxDim[X, i]$ - za $i = 1$ daje maksimalnu dimenziju vrste, a za $i = 2$ maksimalnu dimenziju kolone date matrice X;

```
MaxDim[a_List,broj_]:=
   Block[{A=a,l={}},
     For[i=1,i<=Length[A],i++,
        l=Append[l,A[[i,broj]]]];
     Max[l]];
```

$SS[X, i, j]$ - daje vrednost elementa {i,j} date matrice X, kao i njegovo mesto u listi reprezentacije te matrice;

```
SS[a_List,i1_,i2_]:=
   Block[{A=a,nadji=True,i=1},
```

```
While[nadji,
     If[(A[[i,1]]==i1)&&(A[[i,2]]==i2),
       s=A[[i,3]];mesto=i;nadji=False,
       If[i==Length[A],s=0;nadji=False,i++]]];
 {s,mesto}];
```

$Mnz[X, Y]$ - daje proizvod matrica X i Y u obliku liste uređenih trojki;

```
Mnz[a_List,b_List]:=
   Block[{A=a,B=b,C={},Amb={}},
    m1=MaxDim[A,1]; m2=MaxDim[B,1];n1=MaxDim[A,2];
    n2=MaxDim[B,2]; m=Max[m1,m2]; n=Max[n1,n2];
    For[i=1,i<=m,i++,
     For[j=1,j<=n,j++,
      C=Append[C,{i,j,Sum[SS[A,i,k][[1]]*SS[B,k,j][[1]],{k,1,n}]}]]];
    For[i=1,i<=Length[C],i++,
       If[C[[i,3]]=!=0,Amb=Append[Amb,C[[i]]]]];
    Amb];
```

$MnzSk[X, br]$ - daje proizvod matrice X i skalara br u obliku liste uređ enih trojki;

```
MnzSk[a_List,b_]:=
   Block[{A=a},
    For[i=1,i<=Length[a],i++,A[[i,3]]=A[[i,3]]*b];
    A];
```

$Rcpv[X]$ - daje matricu čiji su elementi recipročne vrednosti matrice X;

```
Rcpv[a_List]:=
   Block[{A=a},
     For[i=1,i<=Length[a],i++,A[[i,3]]=1/A[[i,3]]];
   A];
```

$Hermit1[X]$ - daje transponovanu matricu X;

```
Hermit1[a_]:=
   Block[{A=a,t},
     For[i=1,i<=Length[a],i++,
         t=A[[i,1]];A[[i,1]]=a[[i,2]];A[[i,2]]=t];
     A];
```

$Odz[X, Y]$ - oduzima od matrice X matricu Y i daje rezultat u obliku liste uređenih trojki. Funkcija se koristi i za sabiranje dve matrice ako se za matricu Y uzme $Y = MnzSk[Y, -1]$;

```
Odz[a_List,b_List]:=
   Block[{A=a,B=b,g,C={},t,t1},
```

```
        For[i=1,i<=Length[A],i++,
           s1=A[[i,1]]; s2=A[[i,2]]; g=SS[B,s1,s2];
           t=g[[1]]; t1=g[[2]];
           If[t==0,C=Append[C,A[[i]]],
             If[A[[i,3]]=!=t,C=Append[C,{s1,s2,A[[i,3]]-t}]];
             B=Complement[B,{B[[t1]]}]
        ]   ];
        For[i=1,i<=Length[B],i++,
           C=Append[C,{B[[i,1]],B[[i,2]],-B[[i,3]]}]];
        C];
```

$Adop1[X, i]$ - daje matricu koja sadrži prvih i kolona date matrice X.

```
Adop1[a_List,j_]:=
   Block[{A=a,i,C={}},
    For[i=1,i<=Length[A],i++,
       If[A[[i,2]]<=j,C=Append[C,A[[i]]]]];
    C];
```

Primer 2.41 Neka su date matrice:

$$A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & -1 & -3 \end{bmatrix}, \quad Y1 = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 2 & 0 & 1 & 0 \end{bmatrix} \quad i \quad Z1 = \begin{bmatrix} 1 & 3 \\ 0 & 1 \\ 2 & 0 \\ 0 & 0 \end{bmatrix}$$

Matrice $Y1$ i $Z1$ se mogu predstaviti kao liste uređenih trojki respektivno, na sledeći način:

$Y = \{\{1,1,1\},\{1,2,3\},\{1,4,1\},\{2,1,2\},\{2,3,1\}\}$

$Z = \{\{1,1,1\},\{1,2,3\},\{2,2,1\},\{3,1,2\}\}$

In[1]:=Nab[{{-1,0,1,2},{-1,0,0,-1},{0,0,1,3},{0,0,-1,-3}}]
Out[1]={{1,2,3,4},{1,3,4},{{1,1,-1},{1,3,1},{1,4,2},
{2,1,-1},{2,4,-1},{3,3,1},{3,4,3},{4,3,-1},{4,4,-3}}}

In[2]:=UzmiK[Y,1]
Out[2]={{1,1,1},{2,1,2}}

In[3]:=MaxDim[Y,2]
Out[3]= 4

In[4]:=SS[Y,2,3]
Out[4]={1,5}

$$Y1.Z1 = \begin{bmatrix} 1 & 6 \\ 4 & 6 \end{bmatrix} \text{ odnosno } Z1.Y1 = \begin{bmatrix} 7 & 3 & 3 & 1 \\ 2 & 0 & 1 & 0 \\ 2 & 6 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In[5]:=Mnz[Y,Z]
Out[5]={{1,1,1},{1,2,6},{2,1,4},{2,2,6}}

In[6]:=Mnz[Z,Y]
Out[6]={{1,1,7},{1,2,3},{1,3,3},{1,4,1},{2,1,2},{2,3,1},
{3,1,2},{3,2,6},{3,4,2}}

In[7]:=MnzSk[Y,5]
Out[7]={{1,1,5},{1,2,15},{1,4,5},{2,1,10},{2,3,5}}

In[8]:=Rcpv[Y]
Out[8]=$\{\{1,1,1\},\{1,2,\frac{1}{3}\},\{1,4,1\},\{2,1,\frac{1}{2}\},\{2,3,1\}\}$

In[9]:=Hermit1[Y]
Out[9]={{1,1,1},{2,1,3},{4,1,1},{1,2,2},{3,2,1}}

In[10]:=Odz[Y,Z]
Out[10]={{1,4,1},{2,1,2},{2,3,1},{2,2,-1},{3,1,-2}}

In[11]:=Adop1[Y,3]
Out[11]={{1,1,1},{1,2,3},{2,1,2},{2,3,1}}

2.3.1 Metoda pregradjivanja za retkoposednute matrice

Metoda pregrađivanja može se modifikovati za retko posednute matrice, imajući u vidu napred navedene funkcije za rad sa listama uređenih trojki.

Implementacija metoda

```
DD1[a_List,i_]:=DD1[a,i]=
    Block[{s={}},
      s=Mnz[A1[a,i-1],UzmiK[a,i]];
      If[Length[s]==0,Return[0],Return[s]]
];

CC1[a_List,i_]:=CC1[a,i]=
   If[Length[DD1[a,i]]==0,
      Odz[UzmiK[a,i],MnzSk[Adop1[a,i-1],DD1[a,i]]],
      Odz[UzmiK[a,i],Mnz[Adop1[a,i-1],DD1[a,i]]]
];

B1[a_List,i_]:=B1[a,i]=
   Block[{b=a},
   If[Length[CC1[a,i]]==0,
    If[Length[DD1[a,i]]==0,
     MnzSk[A1[a,i-1],DD1[a,i]/(1+DD1[a,i])],
     Mnz[MnzSk[A1[a,i-1],1/(1+Mnz[Hermit1[DD1[a,i]],DD1[a,i]])],
         DD1[a,i]]
    ],
    b=Rcpv[Mnz[Hermit1[CC1[a,i]],CC1[a,i]]];
    If[Length[b]==0,MnzSk[CC1[a,i],b[[1,3]]],Mnz[CC1[a,i],b]]
```

```
      ]
   ];

   A1[a_List,i_]:=A1[a,i]=
      Block[{b=a,h=h1={},j,mx},
       If[i==1,Mnz[Rcpv[Mnz[Hermit1[UzmiK[a,1]],UzmiK[a,1]]],
                   Hermit1[UzmiK[a,1]]],
         If[Length[DD1[a,i]]==0,
           b=Odz[A1[a,i-1],MnzSk[Hermit1[B1[a,i]],DD1[a,i]]],
           b=Odz[A1[a,i-1],Mnz[DD1[a,i],Hermit1[B1[a,i]]]]
         ];
         For[j=1,j<=Length[b],j++,
            If[b[[j,3]]=!=0,h=Append[h,b[[j]]]]
         ];
         b=h; mx=MaxDim[b,1]; h1=Hermit1[B1[a,i]];
         For[j=1,j<=Length[h1],j++,
             h=Append[h,{mx+1,h1[[j,2]],h1[[j,3]]}]
         ];
         b=Union[b,h]]
   ];

   PregradRet[a_List]:=
      Block[{X=a,n,m},
        X=Nab[a][[3]];n=MaxDim[X,2];
        Print["MOORE-PENROSE INVERZ="];
        A1[X,n]
      ]
```

Primer 2.42 Za matricu $X1 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 1 \end{bmatrix}$ ogdgovarajuća reprezentacija u obliku tro-dimenzionalne liste je:

$$X = \{\{1,1,1\},\{1,2,3\},\{2,1,2\},\{2,2,2\},\{3,1,3\},\{3,2,1\}\}.$$

In[1]:=A1[X,1]
Out[1]=$\{\{1,1,\frac{1}{14}\},\{1,2,\frac{1}{7}\},\{1,3,\frac{3}{14}\}\}$
In[2]:=DD1[X,2]
Out[2]=$\frac{5}{7}$
In[3]:=CC1[X,2]
Out[3]=$\{\{1,1,\frac{16}{7}\},\{2,1,\frac{4}{7}\},\{3,1,-\frac{8}{7}\}\}$
In[4]:=A1[X,2]
Out[4]=$\{\{1,1,-\frac{1}{6}\},\{1,2,\frac{1}{12}\},\{1,3,\frac{1}{3}\},\{\{2,1,\frac{1}{3}\},\{2,2,\frac{1}{12}\},\{2,3,-\frac{1}{6}\}\}\}$

Primer 2.43 Za matricu $X = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 3 & 1 \\ 0 & -1 & 0 \end{bmatrix}$ pomoću funkcije $Partitioning[\,]$ dobijamo:

In[5]:=Partitioning[X]

A[X,1]={1,0,0}
DD[X,2]=0
CC[X,2]={{0},{3},{-1}}
B[X,2]=$\{\{0\},\{\frac{3}{10}\},\{-\frac{1}{10}\}\}$
A[X,2]=$\{\{1,0,0\},\{0,\frac{3}{10},-\frac{1}{10}\}\}$
DD[X,3]=$\{\{-2\},\{\frac{3}{10}\}\}$
CC[X,3]=$\{\{0\},\{\frac{1}{10}\},\{\frac{3}{10}\}\}$
B[X,3]={{0},{1},{3}}
A[X,3]={{1,2,6},{0,0,-1},{0,1,3}}

Out[5]=MOORE-PENROSE INVERS={{1,2,6},{0,0,-1},{0,1,3}}

Ako matricu X predstavimo kao listu uređenih trojki imamo:

X={{1,1,1},{1,3,-2},{2,2,3},{2,3,1},{3,2,-1}}

Pomoću funkcije $PregradRet[]$ dobijamo sledeće rezultate:

In[6]:=PregradRet[X]

A1[X,1]={{1,1,1}}
DD1[X,2]=0
CC1[X,2]={{2,1,3},{3,1,-1},{1,1,0}}
B1[X,2]=$\{\{2,1,\frac{3}{10}\},\{3,1,-\frac{1}{10}\}\}$
A1[X,2]=$\{\{1,1,1\},\{2,2,\frac{3}{10}\},\{2,3,-\frac{1}{10}\}\}$
DD1[X,3]=$\{\{1,1,-2\},\{2,1,\frac{3}{10}\}\}$
CC1[X,3]=$\{\{2,1,\frac{1}{10}\},\{3,1,\frac{3}{10}\}\}$
B1[X,3]={{2,1,1},{3,1,3}}
A1[X,3]={{1,1,1},{1,2,2},{1,3,6},{2,3,-1},{3,2,1},{3,3,3}}

Out[6]=MOORE-PENROSE INVERS={{1,2,6},{0,0,-1},{0,1,3}}

2.3.2 Determinantska reprezentacija i retke matrice

Algoritam za izračunavanje Moore-Penroseovog inverza implementiran funkcijom $MPInverse[\,]$ opisan je u ranije. Ovde ćemo opisati modifikaciju ovog algoritma za retko posednute matrice. Naime, ako je veliki broj elemenata matrice jednak nuli, onda je izračunavanje svih elemenata inverza nepotrebno.

Funkcija $Nab[\,]$ kao izlaz daje tri liste. Prva lista je sa elementima za α a druga za β, po kojima se trebaju vršiti izračunavanja. u algoritmu za funkciju $MPInverse[\,]$. Bez ove funkcije izračunavali bi se svi elementi za $\alpha \in \{1, m\}$ a $\beta \in \{1, n\}$.

Primer 2.44 Za matricu

$$X = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & -1 & -3 \end{bmatrix}$$

pozivom funkcije $Nab[]$ dobijamo:
In[1]:=Nab[X][[1]]
Out[1]={1,2,3,4}
In[2]:=Nab[X][[2]]
Out[2]={1,3,4}

Koristeći funkciju $Nab[\,]$, modifikacija za $MPInverse[\,]$ se može izvršiti na sledeći način:

```
MPInvN[a_]:=
   Block[{b=a,c,p,q,ff={},ra,m,n, d,w,v,i,j,k, j1, p1,q1,
    pr,pr1, awv,mr,mc,s,inv,am},
    inv=Transpose[b]; ra=rank[b]; d=Dimensions[b];
    m=d[[1]]; n=d[[2]]; d=GDet1[b]; p=Table[k,{k,ra}];
    q=Table[k,{k,ra}]; ff=Nab[b]; pp=ff[[1]]; qq=ff[[2]];
    Print["pp=",pp]; Print["qq=",qq]; q1=q; p1=p;
    For[v=1, v<=n, v++,
      For[w=1, w<=m, w++,
         If[MemberQ[qq,v] && MemberQ[pp,w],
             s=0;
             If[ra==m, j=1, j=m];
               While[j>=1, If[ra==n, j1=1, j1=n];
                  While[j1>=1,pr=1; pr1=1;
                    While[pr<=ra && p[[pr]]=!=w, pr++];
                     While[pr1<=ra && q[[pr1]]=!=v, pr1++];
                       If[pr<=ra && pr1<=ra,
                         mr=Minor1[b,p,q,ra]; mc=Det[mr];
                         am=Det[MatrixComp[mr,pr,pr1]];
                         awv=(-1)^(pr+pr1) am mc,
                         awv=0
                       ];
                       s+=awv;
                       If[q[[ra]]==n, j1--, j1=ra ];
                       If[j1>=1,For[i=ra, i>=j1, i--,
                               q[[i]]=q[[j1]]+i-j1+1;
                               q1[[i]]=q[[i]]
```

```
                    ]   ]   ];
                     If[ra<n, j1=n; q=Table[k,{k,ra}]; q1=q, j1=0];
                     If[p[[ra]]==m, j--, j=ra ];
                     If[j>=1,
                         For[i=ra, i>=j, i--,
                               p[[i]]=p[[j]]+i-j+1; p1[[i]]=p[[i]]
                     ]       ];
                     If[ra==m,j=0]
                    ];
                inv[[v,w]]=s/d; p=Table[k,{k,ra}]; q=Table[k,{k,ra}];
                q1=q; p1=p]
            ] ]];
          MatrixForm[Simplify[inv]]
       ]/; MatrixQ[a]
```

Algoritam se razlikuje od osnovnog $MPInverse[\,]$, u delu u kome se izbacuju izračunavanja minora koji su sigurno jednaki nula.

```
ff=Nab[b]; pp=ff[[1]]; qq=ff[[2]]; q1=q; p1=p;
For[v=1, v<=n, v++,
  For[w=1, w<=m, w++,
     If[MemberQ[qq,v] && MemberQ[pp,w],
...  ...
]]     ]
```

Ovakva modifikacija je posebno značajna za izračunavanje inverza retko posednudih matrica jer se višestruko skraćuje izračunavanje nepotrebnih elemenata.

Primer 2.45 Za retko posednutu matricu $A = \begin{bmatrix} -1 & 0 & 3 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$ pozivom funkcije $MPInvN[\,]$ dobijamo rezultat za 0.77 sekunde.

In[2]:=MPInvN[{{-1,0,3,0},{-1,0,0,0},{0,0,1,0},{0,0,-1,0}}]

l1={{1,1,-1},{1,3,3},{2,1,-1},{3,3,1},{4,3,-1}}
pp={1,2,3,4}
qq={1,3}

Out[2]= $\begin{bmatrix} -\frac{2}{13} & -\frac{11}{13} & \frac{3}{13} & -\frac{3}{13} \\ 0 & 0 & 0 & 0 \\ \frac{3}{13} & -\frac{3}{13} & \frac{2}{13} & -\frac{2}{13} \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Za istu matricu A primenom funkcije $MPInverze[]$ opisane ranije, rezultat se dobija za 1.27 sekunde.

Razlika u vremenu izračunavanja inverza se drastično povećava sa povećanjem dimenzija matrice.

In[3]:=MPInverse[{{-1,0,3,0},{-1,0,0,0},{0,0,1,0},{0,0,-1,0}}]

Rezultat se dobija za 1.27 sekunde.

$$\text{Out[3]}=\begin{bmatrix} -\frac{2}{13} & -\frac{11}{13} & \frac{3}{13} & -\frac{3}{13} \\ 0 & 0 & 0 & 0 \\ \frac{3}{13} & -\frac{3}{13} & \frac{2}{13} & -\frac{2}{13} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Primer 2.46 Za matricu A datu sa:

In[4]:=MPInvN[{{-1,0,3,0},{0,0,-1,0},{0,0,2,0},{0,0,0,0}}]

l1={{1,1,-1},{1,3,3},{2,3,-1},{3,3,2}}

pp={1,2,3}

qq={1,3}

Rezultat se dobija za 0.61 sekunde.

$$\text{Out[4]}=\begin{bmatrix} -1 & -\frac{3}{5} & \frac{6}{5} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{5} & \frac{2}{5} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In[5]:=MPInverse1[{{-1,0,3,0},{0,0,-1,0},{0,0,2,0},{0,0,0,0}}]

Rezultat se dobija za 1.32 sekunde,

$$\text{Out[5]}=\begin{bmatrix} -1 & -\frac{3}{5} & \frac{6}{5} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{5} & \frac{2}{5} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Algoritam za $R-inverza$, funkcijom $RInverse[\,]$, se takođe može modifikovati u istom delu kao i algoritam za $MPInverse[\,]$. Naravno, ovde je potrebno obraditi matrice A i R, pa se algoritam malo razlikuje. Kada su pronađeni α_A, α_R, β_A i β_R, potrebno je naći α kao presek α_A i α_R, i β kao presek β_A i β_R. Ovo proizilazi iz činjenice da ukoliko se neki element iz α_A ne nalazi u α_R njega ne treba uzimati pri izračunavanju. Isto važi i za β_A i β_R. Na ovaj način se, višestruko skraćuje izračunavanje R-inverza.

Deo u kome se modifikacija razlikuje od implementirane funkcije $RInverse[\,]$ je sledeći:

```
ff=Nab[b];pp1=ff[[1]]; qq1=ff[[2]]; ff={}; ff=Nab[t,m,n];
pp2=ff[[1]]; qq2=ff[[2]];pp=Intersection[pp1,pp2];
qq=Intersection[qq1,qq2];
For[v=1, v<=n, v++,
```

```
    For[w=1, w<=m, w++,
       If[MemberQ[qq,v] && MemberQ[pp,w],
  ...  ...
  ]]     ]
```

Modifikovana funkcija je $RInvN[\,]$.

Primer 2.47 Za retko posednute matrice

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \text{ i } R = \begin{bmatrix} -1 & 0 & 2 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

pozivom funkcije $RInverze[]$ dobijamo rezultat za 1.65 sekunde.

In[6]:=RInverse[{{-1,0,0,0},{0,0,-1,0},{0,0,2,0},{1,0,0,0}},
{{-1,0,2,0},{0,0,-1,0},{1,0,2,0},{1,0,0,0}}]

Out[6]= $\begin{bmatrix} -\frac{9}{14} & \frac{1}{7} & \frac{1}{14} & \frac{5}{14} \\ 0 & 0 & 0 & 0 \\ \frac{1}{7} & -\frac{1}{7} & \frac{3}{7} & \frac{1}{7} \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Dok se modifikovanim algoritmom rezultat dobija za upola kraće vreme.

In[7]:=RInvN[{{-1,0,0,0},{0,0,-1,0},{0,0,2,0},{1,0,0,0}},
{{-1,0,2,0},{0,0,-1,0},{1,0,2,0},{1,0,0,0}}]]

l1={{1,1,-1},{2,3,-1},{3,3,2},{4,1,1}}

l1={{1,1,-1},{1,3,2},{2,3,-1},{3,1,1},{3,3,2},{4,1,1}}

pp={1,2,3,4}

qq={1,3}

Rezultat se dobija za 0.85 sekunde,

Out[7]= $\begin{bmatrix} -\frac{9}{14} & \frac{1}{7} & \frac{1}{14} & \frac{5}{14} \\ 0 & 0 & 0 & 0 \\ \frac{1}{7} & -\frac{1}{7} & \frac{3}{7} & \frac{1}{7} \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Za merenje rezultata korišćena je funkcija $Timing[\,]$.

3 GENERALISANI INVERZI I BAZE PODATAKA

Pristup kojim se izbegavaju ponovna izračunavanja nekih vrednosti, pri rekurzivnim pozivima, bazira se na konstrukciji nekoliko baza podataka. Potrebne vrednosti izračunavaju se samo jednom i smestaju u određenu bazu. U izrazu koji koristi ranije izračunate vrednosti potrebno je samo pročitati vrednost iz odgovarajuće baze podataka u koju je smešten [74, 80].

Za manipulaciju bazama podataka u tezi je korišćen programski paket DELPHI 6.0. Takođe, je korišćen DATABASE DESKTOP 7.0 FOR WINDOWS 95 & WINDOWS NT za pravljenje potrebnih baza [37, 11, 12].

3.1 METODA DETERMINANTSKE REPREZENTACIJE

Metoda determinantske reprezentacije vodi ponovnom izračunavanju istih minora za dve date matrice. Tim pre, algoritam zahteva izračunavanja značajnog broja različitih minora. Pomoću baza podataka, potrebni minori izračunavaju se samo jednom i memorišu. U izrazu koji koristi ranije izračunate minore potrebno je samo pročitati njegovu vrednost iz odgovarajuće baze podataka u koju je smešten [74].

Algoritam 3.1 *Neka su A i R dim $m \times n$ kompleksne matrice ranga r. Ako skup $\mathcal{Q}_{t,m} \times \mathcal{Q}_{t,n}$, $t \leq r = rank(A)$ označimo sa $\mathcal{N}(t)$, onda generalizovanu determinantu reda t, označavamo sa $\mathrm{DET}_{(\mathrm{R,t})}(\mathrm{A})$, možemo pisati na sledeći način:*

$$\mathrm{DET}_{(\mathrm{R,t})}(\mathrm{A}) = \sum_{(\gamma,\delta)\in\mathcal{N}(\mathrm{t})} [\overline{\mathrm{R}}_{\delta}^{\gamma}][\mathrm{A}_{\delta}^{\gamma}]. \tag{3.1.1}$$

Ako uvedemo sledeća označavanja:

$$\mathcal{I}(\alpha,t) = \{I:\ I \in \mathcal{Q}_{t,m},\ I \supseteq \alpha\}, \quad \mathcal{J}(\beta,t) = \{J:\ J \in \mathcal{Q}_{t,n},\ J \supseteq \beta\},$$

$$\mathcal{N}(\alpha,\beta,t) = \mathcal{I}(\alpha,t) \times \mathcal{J}(\beta,t), \quad t \leq r,$$

onda generalna determinantska reprezentacija reda t za $A \in \mathbb{C}_r^{m\times n}$, može biti napisana kao:

$$g_{ij} = \frac{\sum\limits_{(\alpha,\beta)\in\mathcal{N}(j,i,t)} [\overline{R}_{\beta}^{\alpha}] \frac{\partial}{\partial a_{ji}} [A_{\beta}^{\alpha}]}{\mathrm{DET}_{(\mathrm{R,t})}(\mathrm{A})}, \quad \begin{pmatrix} 1 \leq i \leq n \\ 1 \leq j \leq m \end{pmatrix}. \tag{3.1.2}$$

Definicije generalizovane determinante i generalne determinantske reprezentacije različitog reda su korisne u slučaju kad je rang date matrice A nepoznat.Izračunavanja koja počinju sa (3.1.2) gde je $t = \min\{m, n\}$ i prvi ceo integer t koji zadovoljava $\mathrm{DET}_{(\mathrm{R,t})}(\mathrm{A}) \neq 0$ jednak je rangu $\mathrm{rank}(A)$.

Implementacija algoritma

Algoritam zahteva veliki broj izračunavanja različitih minora. Ako su matrice A i R reda $m \times n$ i ranga r, izrazi (3.1.1) i (3.1.2) zahtevaju $\binom{m}{r} \times \binom{n}{r}$ različitih minora matrice A kao i $\binom{m}{r} \times \binom{n}{r}$ različitih minora matrice R. Čak šta više izraz (3.1.2) zahteva $\binom{m-1}{r-1} \times \binom{n-1}{r-1}$ različitih minora matrice A.

Pri implementaciji algoritma u proceduralnim programskim jezicima glavni problem je tzv. memory overflow. Ako matrice A, R, G deklarišemo kao dvodimenzionalne nizove, onda su dimenzije matrica A, R, G ograničene unapred. Kada su matrice A, R, G implementirane kao dinamičke strukture, povezane liste, mnogo memoriskog prostara se zauzima, i nastaje tzv. stack overflow problem. Kao rešenje ovog problema preporučena je manipulacija bazama podataka. Glavna ideja je ta da se matrice i svi rezultati smeste u odgovarajuću bazu, umesto u memorijske promenljive. Koristeći baze, u mogućnosti smo da rešimo dva velika problema: ponovno izračunavanje minora nižeg reda kao i ponovno izračunavanje istih minora [74].

Glavna ideja koja je korist za realizaciju ovog pristupa je opisana u sledećim koracima:

Korak 1. Inicijalne vrednosti. Izračunavamo 2×2 minore matrica A i R, i smeštamo vrednosti u bazu.

Korak 2. Rekurzivni korak. Svaki minor reda $d \times d$ može biti izračunat koristeći razvoj po njegovoj prvoj vrsti i smeštenim vrednostima za $(d-1)\times(d-1)$ minore u bazi. Tražena vrednost minora reda $(d-1) \times (d-1)$ se ne izračunava svaki put kad nam treba, već se jednostavno uzme iz baze.

Korak 3. Zaustavni kriterijum. Proces prekidamo u slučaju kad su svi smešteni minori identički jednaki nuli.

Mogućnost ove implementacije je da izbegne ponovo izračunavanje istih minora i sadržna je u koraku 2. Bilo koja tražena vrednost za $(d-1) \times (d-1)$ minore se ne izračunava već se jednostavo uzima iz baze.

Zaustavni kriterijum opisan u koraku 3 daje mogućnost za eksplicito nalaženje ranga date matrice A. Preciznije, u slučaju kad su sve smeštene vrednosti minora reda $d \times d$ jednake nuli, dat je $\operatorname{rank}(A) = d - 1$. Ovo je jedna od korisnih osobina "dobrih" numeričkih algoritama za izračunavanje generalisanih inverza, predloženih u [46].

Svaki minor reda 2×2 može biti izračunat pomoću 3 operacija. Skup svih 2×2 minora matrice A i R može se formirati u

$$2\begin{pmatrix} m \\ 2 \end{pmatrix}\begin{pmatrix} n \\ 2 \end{pmatrix}3$$

operacija. Svaki minor reda $d \geq 3$ može se izračunati sa samo d množenja. Skup svih $d \times d$ minora za A i R $(d \geq 3)$ je formiran uz pomoć

$$2\begin{pmatrix} m \\ d \end{pmatrix}\begin{pmatrix} n \\ d \end{pmatrix}d$$

množenja. Ukupan broj operacija za sve $r \times r$ minore za A i R je

$$\sum_{d=3}^{r} 2\begin{pmatrix} m \\ d \end{pmatrix}\begin{pmatrix} n \\ d \end{pmatrix} \cdot d + 2\begin{pmatrix} m \\ 2 \end{pmatrix}\begin{pmatrix} n \\ 2 \end{pmatrix} \cdot 3.$$

Na kraju, izračunavanje traženog g-inverse saglasno (4.2) zahteva

$$mn\left(\sum_{d=3}^{r} 2\begin{pmatrix} m \\ d \end{pmatrix}\begin{pmatrix} n \\ d \end{pmatrix} \cdot d + 2\begin{pmatrix} m \\ 2 \end{pmatrix}\begin{pmatrix} n \\ 2 \end{pmatrix} \cdot 3\right) \cdot 2$$

operacija.

U implementaciji se koriste šest baza: MATRICA.DB, MINORI.DB, PREMINORI.DB i TRENUTNA.DB i definisane su na sledeći način:

	Field_Name	**Type**	**Size**	**Key**
1	Vr_kol	A	10	$*$
2	$Vrednost$	N		
3	$Vredn1\_R$	N		

Simbol $*$ je ključ za pretraživanje u ovim bazama. Slovo A označava alfanumeričko polje, a slovo N označava numeričko polje.

Baza GIJB.DB ima oblik:

	Field_Name	**Type**	**Size**
1	Vr_kol	A	10
2	Dim	N	
3	$Vrednost$	N	

Baza KOMBIN.DB ima oblik:

	Field_Name	**Type**	**Size**	**Key**
1	*Dim*	*A*	3	$*$
2	*Rbr*	*A*	7	$*$
3	*Tip*	*A*	1	$*$
4	*Kom*	*A*	255	

Primarni indeks pretraživanja je *Key*=”Dim;Rbr;Tip”, dok je sekundarni indeks pretraživanja *Key*=”Dim;Tip;Kom”. Tip polje ima dve vrednosti ”v” ili ”k”, saglasno tome dali se radi o kombinaciji za vrstu ili kolonu.

KOMBIN.DB, sa oznakom *Table1*: je baza u kojoj se smeštaju kombinacije vrsta i kolona,

MATRICA.DB, sa oznakom *Table2*: je baza za elemente matrica A i R,

TRENUTNA.DB, sa oznakom *Table3*: je baza za trenutno izra vcunate vrednosti,

MINORI.DB, sa oznakom *Table4*: je baza za minore reda $d-1$,

PREDMINORI.DB, sa oznakom *Table5*: je baza za minore reda $d-2$,

GIJB.DB, sa oznakom *Table6*: je baza u kojoj se smeštaju elementi rezultujuće matrice G.

U svakoj itraciji sadržaj iz *Table4* se kopira u *Table5*, a sadržaj iz *Table3* se kopira u *Table4*, dok se sadržaj *Table5* briše. Kombinacije u bazi *Table1* su smeštene kao stringovi odvojeni simbolom različitim od decimalne cifre $0,\dots,9$. Na primer, korišćen je simbol ’,’.

Primer 3.1 Matrice

$$A=\begin{bmatrix} 0 & 1 & -1 & 0\\ 2 & 0 & 7 & 0\\ 1 & 1 & -1 & 1\end{bmatrix};\ R=\begin{bmatrix} 1 & -3 & 2 & 0\\ 0 & 0 & -2 & 1\\ 1 & 0 & -1 & 0\end{bmatrix}$$

su smeštene u bazi *MATRICA.DB* na sledeći način:

Vr_kol	**Vrednost**	**Vredn1_R**
1/1	0	1
1/2	1	−3
1/3	−1	2
1/4	0	0
2/1	2	0
2/2	0	0
2/3	7	−2
2/4	0	1
3/1	1	1
3/2	1	0
3/3	−1	−1
3/4	1	0

Primer 3.2 Za matrice A i R date u Primeru 4.1, baza $KOMBIN.DB$ sadrži sledeće podatke:

Dim	**Rbr**	**Tip**	**Kom**
2	1	k	$1,2$
2	1	v	$1,2$
2	2	k	$1,3$
2	2	v	$1,3$
2	3	k	$1,4$
2	3	v	$2,3$
2	4	k	$2,3$
2	5	k	$2,4$
2	6	k	$3,4$
3	1	k	$1,2,3$
3	1	v	$1,2,3$
3	2	k	$1,2,4$
3	3	k	$1,3,4$
3	4	k	$2,3,4$

Na primer kada tražimo 2. kombinaciju dimenzije 3 od 4 elementa, po kolonama, onda je ključ pretraživanja baze KOMBIN.DB dat sa:

Key="3;2;k", a rezultat je '1,2,4'.

Ako bi recimo tražili redni broj kombinacije po vrsti '2,3', kjuč pretraživanja baze bio bi:

Key="2;v;2,3", a rezultat je 3.

Sledi izvorni kod funkcije *gener*, kojom se u bazu KOMBIN.DB, upisuju dimenzija, redni broj, tip i izgled kombinacije. Za vrstu je parametar baz="v", a za kolonu baz="k".

```
function TDetForm.gener(k,n:Integer;baz:String):integer;
type TNiz=Array[1..90] of Integer;
var  c:Tniz; z,i,brk:Integer;
   k1,br1,pr1,prm,ddim,bbrk:string;
begin
   If k<n then
   begin
   for I:=1 to k do c[I]:=I;
   z:=k; brk:=0;
   while z≥1 do
    begin
     brk:=brk+1; prm:='';
     for I:=1 to k-1 do prm:=prm+IntToStr(c[I])+',';
     prm:=prm+IntToStr(c[k]); ddim:=IntToStr(k); bbrk:=IntToStr(brk);
     with Table1 do
```

```
      begin
        append;
        fieldbyname('dim').asstring:=ddim; fieldbyname('rbr').asstring:=bbrk;
        fieldbyname('kom').asstring:=prm;
        if baz='k' then fieldbyname('tip').asstring:='k';
           else fieldbyname('tip').asstring:='v';
      end;
      if c[k]=n then z:=z-1 else z:=k;
      if z≥1 then for I:=k downto z do c[I]:=c[z]+I-z+1;
     end; end
   else begin
     brk:=1; prm:='';
     for I:=1 to k-1 do prm:=prm+IntToStr(I)+',';
     prm:=prm+IntToStr(k); ddim:=IntToStr(k); bbrk:=IntToStr(brk);
     with Table1 do
     begin
        append;
        fieldbyname('dim').asstring:=ddim; fieldbyname('rbr').asstring:=bbrk;
        fieldbyname('kom').asstring:=prm;
        if baz='k' then fieldbyname('tip').asstring:='k';
           else fieldbyname('tip').asstring:='v';
     end; end;
   generisi:=brk;
   end;
end;
```

Minori $|A^\alpha_\beta|$ i $|R^\alpha_\beta|$, $(\alpha,\beta)\in\mathcal{N}_d$ reda d se izračunavaju u proceduri *calc*, koristeći prethodno izračunate minore reda $d-1$, koji su smešteni u bazi MINORI.DB. Vrednost minora reda $d-1$ uzima se iz baze MINORI.DB u proceduri *readBaz*. Procedura *readBaz* uzima sledeće formalne parametre:

k1,k2:string; - parametri koji sadrže kombinaciju vrsta i kolona, respektivno, u zahtevanom minoru reda $d-1$.

dd:integer; - parametar reprezentuje stvarnu dimenziju (jednaku $d-1$) prethodno izračunatog minora.

var mt,mt1:real; - ovi izlazni parametri predstavljaju vrednosti minora reda $d-1$, koji su uzeti iz baze MINORI.DB, (*Table4*).

```
procedure TDetForm.readBaz(k1,k2:string;dd:integer; var mt,mt1:real);
var i,j,fd,gde:string;
begin
   if dd=1 then gde:=k1+'/'+k2
   else begin
```

```
      fd:=IntToStr(dd); table1.indexfieldnames:='Dim;Tip;Kom';
      Table1.FindKey([fd,'v',k1]);
      i:=Table1.Fields[1].AsString;
      Table1.FindKey([fd,'k',k2]);
      j:=Table1.Fields[1].AsString;
      gde:=i+'/'+j; table1.indexfieldnames:='Dim;Rbr;Tip';
    end;
    Table4.Open; Table4.FindKey([gde]);
    mt:=Table4.Fields[1].AsFloat; mt1:=Table4.Fields[2].AsFloat;
end;
```

```
procedure TDetForm.pozicija(rec,podrec:string; var element:integer);
var d1,d2,brojac:integer;
   j1,v1,j2,v2:string;
   rad,dobro:boolean;
begin
   d1:=length(rec);d2:=length(podrec);j1:=copy(podrec,1,1);
   j2:=copy(podrec,2,d2-1); brojac:=1; dobro:=false; rad:=true;
   element:=1;
   while rad do
   begin
     v1:=copy(rec,brojac,1); v2:=copy(rec,brojac+1,d2-1);
     if v1=',' then element:=element+1
               else if j1=v1 then
                    if j2=v2 then begin
                                   rad:=False; dobro:=true;
                                  end;
     brojac:=brojac+1;
     if brojac>d1 then rad:=false
   end;
   if not dobro then element:=0;
end;
```

Procedura *pozicija('1,2,10,11,12,13','11',element)* kao rezultat vraća promenljivu *elemetn*=4, tj. poziciju stringa '11' u stringu '1,2,10,11,12,13'.

```
procedure TDetForm.take(rec:string;br:integer;
                    var o1,recbezo1:string);
var d1,u,brojac:integer;
   slovo:string;
   rad:boolean;
begin
   d1:=length(rec); u:=0; o1:=''; recbezo1:='';
```

```
    brojac:=1;
    while u<>br-1 do
    begin
      slovo:=copy(rec,brojac,1);
      if slovo=',' then u:=u+1; recbezo1:=recbezo1+slovo;
      brojac:=brojac+1
    end;
    slovo:=copy(rec,brojac,1); rad:=true;
    while rad do
    begin
      o1:=o1+slovo; brojac:=brojac+1;
      slovo:=copy(rec,brojac,1);
      if slovo=',' then rad:=false;
      if slovo='' then rad:=false;
    end;
    if brojac>=d1
      then recbezo1:=copy(recbezo1,1,length(recbezo1)-1)
      else recbezo1:=recbezo1+copy(rec,brojac+1,d1-brojac)
end;
```

Procedura *take('1,2,10,11,12,13',4,o1,recbezo1)* kao rezultat vraća string *o1*='11' i string *recbezo1*='1,2,10,12,13'.

U proceduri *calc* su korišćeni sledeći formalni parametri:
i,j:integer; - redni broj kombinacije vrsta i kolona, respektivno.
d:integer; - red minora čija se vrednost izračunava.
var s,s1:real; - vrednost rezultujućeg minora reda d, koji odgovara A i R, respektivno.

Ako su redni brojevi kombinacija α i β reda d jednaki I i J, respektivno, onda pišemo $|A^I_J|_d$ umesto $|A^\alpha_\beta|$.

```
procedure TDetForm.calc(i,j,d:integer;var s,s1:real);
var t,znak,poc,d1:integer;Rij,Aij,mt,mt1:real;
   ist,jst,dst,slovo,vr,kol,a1,ost,a2,tt,zk:string;
begin
   s:=0; s1:=0; ist:=IntToStr(i); jst:=IntToStr(j);
   dst:=IntToStr(d); Table1.FindKey([dst,ist,'v']);
   vr:=Table1.Fields[3].AsString;
   Table1.FindKey([dst,jst.'k']);
   kol:=Table1.Fields[3].AsString;
   d1:=length(vr); a1:=''; poc:=0;
   while slovo <> ',' do
   begin
```

```
      poc:=poc+1; a1:=a1+copy(vr,poc,1);
      slovo:=copy(vr,poc+1,1)
    end;
    poc:=poc+1; ost:=copy(vr,poc+1,d1-poc); znak:=1;
    for t:=1 to d do
    begin
     take(kol,t,a2,zk); tt:=a1+'/'+a2; Table2.FindKey([tt]);
     Aij:=Table2.Fields[1].AsFloat;
     Rij:=Table2.Fields[2].AsFloat;
     readBaz(ost,zk,d-1,mt,mt1);
     s:=s+znak*Aij*mt;s1:=s1+znak*Rij*mt1;znak:=(-1)*znak;
    end;
    Table3.Open;
    with Table3 do
    begin
      append;
      fieldbyname('vr_kol').AsString:=ist+'/'+jst;
      fieldbyname('vrednost').AsFloat:=s;
      fieldbyname('vredn1_R').AsFloat:=s1;
      refresh;
    end;
  end;
```

Primer 3.3 Razmotrimo matrice A i R iz prethodnog primera. Njeni minori $|A_3^1|_3$ i $|R_3^1|_3$ su izračunati simultano u proceduri *calc(1,3,3,mt,mt1)*. U izlaznu promenljivu mt je upisana vrednost za traženi minor A, a u promenljivu *mt1* je upisana vrednost za R. Računanjem minora $|A_3^1|_3 = A\begin{bmatrix} 1,2,3 \\ 1,3,4 \end{bmatrix}_3$, koristeći razvoj po prvoj vrsti ovog minora, dobijamo:

$$|A_3^1|_3 \;\; = 1*0*A\begin{bmatrix} 2,3 \\ 3,4 \end{bmatrix}_2 + (-1)*(-1)*A\begin{bmatrix} 2,3 \\ 1,4 \end{bmatrix}_2 + 1*0*A\begin{bmatrix} 2,3 \\ 1,3 \end{bmatrix}_2.$$

Umesto ponovnog računanja minora $A\begin{bmatrix} 2,3 \\ 3,4 \end{bmatrix}_2$ dovoljno je samo pročitati vrednost iz baze MINORI.DB.

To je urađeno pozivom procedure *readBaz('2,3','3,4',2,t,t1)*.

Odgovarajući minor u bazi MINORI.DB traži se pomoću ključa *Key*="3/6", jer je $'2,3'$ treća kombinacija vrsta matrice A reda 2, i $'3,4'$ je šesta kombinacija reda 2 kolona matrice A. Dobijamo $mt = 7$, a $mt1 = 1$ i imamo:

$$A\begin{bmatrix} 2,3 \\ 3,4 \end{bmatrix}_2 = Key(3/6) = 7$$
$$A\begin{bmatrix} 2,3 \\ 1,4 \end{bmatrix}_2 = Key(3/3) = 2$$
$$A\begin{bmatrix} 2,3 \\ 1,3 \end{bmatrix}_2 = Key(3/2) = 0$$

$$|A_3^1|_3 = A\begin{bmatrix} 1,2,3 \\ 1,3,4 \end{bmatrix}_3 = 2$$

U tom slučaju će baza MINORI.DB sadržati sledeće podatke:

Vr_kol	**Vrednost**	**Vredn1_R**
1/1	7	6
1/2	−2	−3
1/3	2	3
1/4	7	−3

Prilikom izračunavanja minora matrica A i R eksplicitno određujemo da je rank(A) jednak rank(R), koristeći sledeću ideju:

- u slučaju kad su svi elementi u bazi TRENUTNA.DB jednaki nuli, zaključujemo da je $r = \text{rank}(A) = \text{rank}(R) = d - 1$. U tom slučaju, u bazi MINORI.DB su smeštene vrednosti mimora reda r, izabranih matrica A i R.

Procedura *GIJCalc* smešta rezultat u bazu GIJB.DB (*Table 6.*) matrice G, izračunat saglasno (3.2.2). Pretpostavimo da su I, J redni redni brojevi kombinacije α, β, respektivno, i da su K, L redni brojevi kombinacije $\alpha\backslash\{j\}$ i $\beta\backslash\{i\}$, respektivno. Tada se vrednost izraza (4.2) može napisati na sledeći način:

$$g_{i,j} = \frac{1}{detRA}\sum_{I,J}|R_J^I|_d(-1)^{\text{poz}}(j,\alpha)+\text{poz}(i,\beta)|A_L^K|_{d-1}$$

Kombinacije I i J su generisane iz KONBIN.DB ključem *Key*="d;I;'v'", odnosno *Key*="d;J;'k'".

U funkciji MAK su izračunate vrednosti algebarskog komplementa

$$\frac{\partial}{\partial a_{ji}}[A_\beta^\alpha], \quad i = 1,\dots,m, \; j = 1,\dots,n.$$

Formalni parametri koji su korišćeni u ovoj funkciji su:

vik : *string*; - string oblika $'i/j'$, gde je i redni broj vrste, a j redni broj kolone.

d : *integer*; - dimenzija podmatrice A_β^α.

```
function TDetForm.MAK(vik:string;j,i,r:integer):real;
var cr,ii,jj:integer;
   ist,jst,rst,vr,kol,d_j,b_i,ipre,jpre,
   m1,gde1:string;ddd:real;
begin
   cr:=pos('/',vik);ist:=copy(vik,1,cr-1);
   jst:=copy(vik,cr+1,r-cr); rst:=IntToStr(r);
   Table1.FindKey([rst,ist,'v']);
   vr:=Table1.fields[3].AsString;
   Table1.FindKey([rst,jst,'k']);
   kol:=Table1.fields[3].AsString; jst:=IntToStr(j);
   cr:=poz(vr,jst);
   if cr=0 then MAK:=0
   else begin
     jj:=cr; take(vr,cr,m1,d_j); ist:=IntToStr(i);
     pozition(kol,ist,cr);
     if cr=0 then MAK:=0
     else begin
      ii:=cr; take(kol,cr,m1,b_i);
      rst:=IntToStr(r-1);
      Table1.Indexfieldnames:='Dim;Tip;Kom';
      Table1.findKey([rst,'v',d_j]);
      ipre:=Table1.fields[1].AsString;
      Table1.FindKey([rst,'k',b_i]);
      jpre:=Table1.Fields[1].AsString;
      gde1:=ipre+'/'+jpre;
      Table1.Indexfieldnames:='Dim;Rbr;Tip';
      Table5.FindKey([gde1]);
      ddd:=Table5.fields[1].AsFloat;
      MAK:=na(ii,jj)*ddd;
     end;
   end;
end;
```

Izračunavanje $\{2\}$ inverza

Elemente matrice G {reda d=2,3,..}, koji ustvari predstavljaju $\{2\}$ inverze nalazimo odmah posle izračunavanja minora datog reda, pozivom procedure *GijCalc*, koja kao parametar ima trenutni red minora u bazi MINORI.DB [74]. Saglasno formuli (3.1.2) može se generisati sledeća procedura kojom se računaju $\{2\}$ inverzi:

```
procedure TDetForm.GijCalc(dd:integer);
var detRA,s,GIJ:real;
   i,j,Ndim,Mdim,rang:integer;gde:string;
begin
   detRA:=0;
   with Table4 do
   begin
    first;
    while not(eof) do
    begin
      detRA:=detRA+fields[1].AsFloat*fields[2].AsFloat;
      next;
    end;
   end;
   table4.open; rang:=StrToInt(rang.text);
   Ndim:=StrToInt(NPolje.text); Mdim:=StrToInt(MPolje.text);
   for i:=1 to Mdim do
   for j:=1 to Ndim do
   begin
     s:=0;
     with Table4 do
     begin
       first;
       while not eof do
       begin
        gde:=Fields[0].AsString; s:=s+Fields[2].AsFloat*MAK(gde,j,i,rang);
        next;
       end;
     end;
   Gij:=s/detRA;
   gde:=inttostr(i)+'/'+inttostr(j);
   with Table6 do
   begin
    append;
    fieldbyname('vr_kol').AsString:=gde;fieldbyname('dim').AsFloat:=rang;
    fieldbyname('vrednost').AsFloat:=Gij;
    refresh;
   end;
   end;
end;
```

Program sadrži sledeće Query-je (upite) za rad sa bazama podataka: *DelTreQuery* je Query za brisanje elemenata iz daze TRENUTNA.DB;

```
DELETE FROM trenutna;
```

InsMinQuery je Query za prepisivanje podataka iz baze TRENUTNA.DB u bazu MINORI.DB;

```
INSERT INTO minori (vr_kol,vrednost,vredn1_R)
SELECT Trenutna.vr_kol, Trenutna.vrednost, Trenutna.vredn1_R
FROM Trenutna;
```

InsPreMinQuety je Query za prepisivanje podataka iz baze MINORI.DB u bazu PREMINORI.DB;

```
INSERT INTO PreMinori (vr_kol,vrednost,vredn1_R )
SELECT Minori.vr_kol, Minori.vrednost, Minori.vredn1_R
FROM Minori.
```

U programu koji je generisan u izvršni fajl *Ginverz.exe* možemo koristiti matrice ranga i do 88. Ovo je jedina restrikcija, koja je prouzrokovana definicijom četvrtog reda baze KOMBIN.DB:

	Field_Name	**Type**	**Size**	**Key**
4	*Kom*	*M*		

Kombinacija $\binom{88}{88} =' 1, 2, 3, \ldots, 86, 87, 88'$ sadrži string dužine 254 znaka!

Ova restrikcija se može izbeći ako polje *Kom* ne bude deklarisano kao alfanumeričko, već kao *Memo* polje. Svaki znak do 254 karaktera u *Memo* polju se smešta u datu bazu, a ostatak se smešta u novu bazu, sa isto deklarisanim *Memo* poljem. Ova baza se automatski vezuje za datu bazu kad se zahteva čitanje znaka posle 254. U ovom slučaju, četvrto polje baze KOMBIN.DB se deklariše kao:

	Field_Name	**Type**	**Size**	**Key**
4	*Kom*	*M*		

gde slovo M označava *Memo* polje.

Primer 3.4 Za matricu, generalisanu zamenom vrednosti $a = 1$ u test matrici preporučenoj u [87] str. 143, naći $\{2\}$ inverze.

Za $A = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 \\ 1 & 3 & 4 & 6 & 2 \\ 2 & 3 & 4 & 5 & 3 \\ 3 & 4 & 5 & 6 & 4 \\ 4 & 5 & 6 & 7 & 6 \\ 6 & 6 & 7 & 7 & 8 \end{bmatrix}$ i $R = \begin{bmatrix} 0 & 1 & 2 & 3 & 0 \\ 0 & 2 & 3 & 5 & 1 \\ 1 & 2 & 3 & 4 & 2 \\ 2 & 3 & 4 & 5 & 3 \\ 3 & 4 & 5 & 6 & 5 \\ 5 & 5 & 6 & 6 & 7 \end{bmatrix}$ rezultat u bazi GIJB, dobijamo sledeće $\{2\}$ inverze:

$$G_2 = \begin{bmatrix} -0.0041 & -0.0005 & -0.0006 & 0.0008 & 0 & 0.0044 \\ -0.0005 & -0.0063 & 0.0005 & 0.0016 & 0.0016 & 0.0032 \\ -0.0006 & 0.0016 & -0.0114 & 0.0025 & 0.0019 & 0.006 \\ 0.0008 & 0.0063 & 0.0034 & -0.0118 & -0.0025 & 0.0038 \\ 0 & 0.0095 & 0.0038 & -0.0038 & -0.0227 & 0.0132 \end{bmatrix}.$$

$$G_3 = \begin{bmatrix} 0.5904 & -0.3142 & -0.1523 & 0.006 & 0.0535 & 0.0911 \\ -0.0076 & -0.0326 & -0.0884 & 0.363 & -0.2434 & -0.0997 \\ 0.1247 & -0.0492 & -0.01 & 0.0605 & -0.014 & 0.0037 \\ -0.003 & 0.115 & 0.0874 & -0.2613 & 0.0236 & 0.0419 \\ -0.4398 & 0.232 & 0.1346 & -0.0771 & 0.1526 & 0.126 \end{bmatrix}.$$

$$G = G_4 = \begin{bmatrix} 0.5 & -0.125 & -1 & 0.875 & -0.625 & 0.375 \\ -1 & 1.875 & -4.5 & 2.875 & -0.625 & 0.375 \\ 1.25 & -1.626 & 3.25 & -1.875 & 0.125 & -0.125 \\ -0.25 & 0.375 & -0.25 & 0.125 & 0.125 & -0.125 \\ -0.5 & -0.25 & 1.5 & -1.25 & 0.75 & -0.25 \end{bmatrix}.$$

Kad je $R = A$, dobijamo Moore-Penrose inverz za A:

$$A^\dagger = \begin{bmatrix} 4 & -1 & -8 & 7 & -5 & 3 \\ -8 & 15 & -36 & 23 & -5 & 3 \\ 10 & -13 & 26 & -15 & 1 & -1 \\ -2 & 3 & -2 & 1 & 1 & -1 \\ -4 & -2 & 12 & -10 & 6 & -2 \end{bmatrix}.$$

Primer 3.5 Za matricu A_{10*11} datu sa:

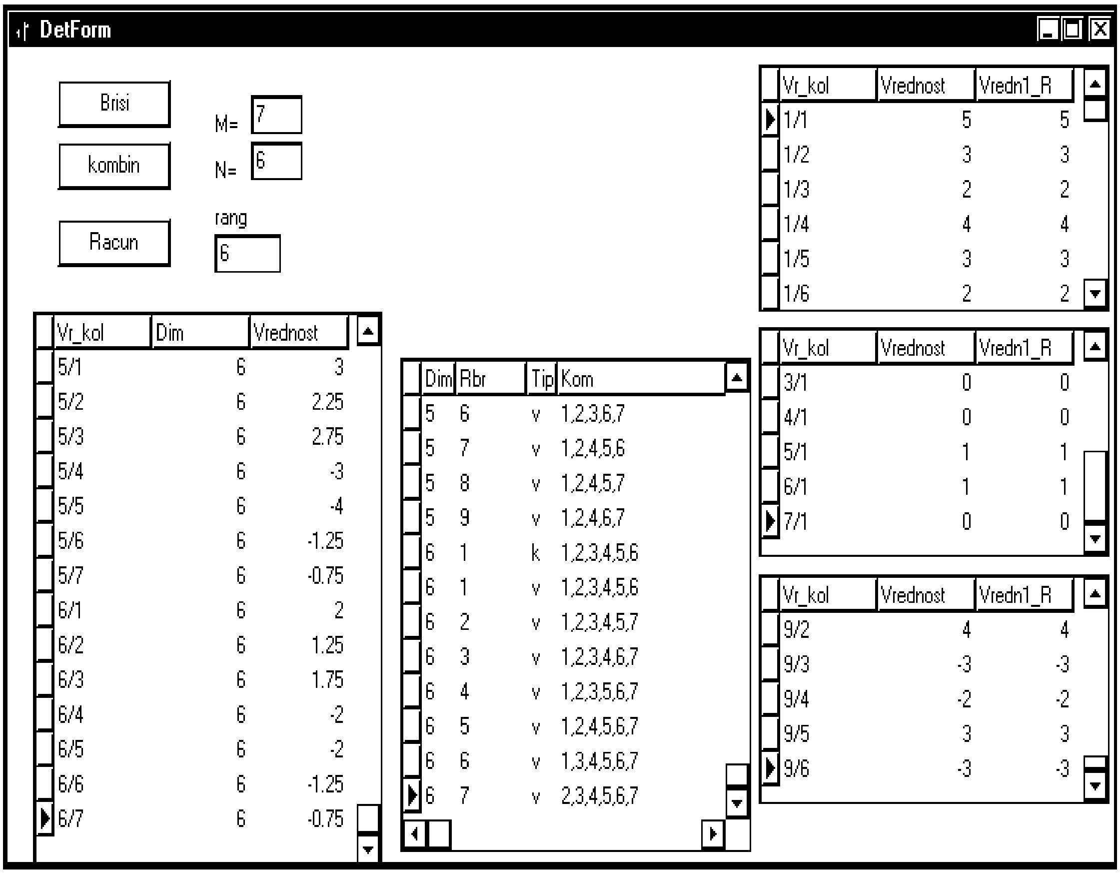
$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 & 1 & 3 & 4 & 6 & 2 \\ 1 & 3 & 4 & 6 & 2 & 2 & 3 & 4 & 5 & 3 \\ 2 & 3 & 4 & 5 & 3 & 3 & 4 & 5 & 6 & 4 \\ 3 & 4 & 5 & 6 & 4 & 4 & 5 & 6 & 7 & 6 \\ 4 & 5 & 6 & 7 & 6 & 6 & 6 & 7 & 7 & 8 \\ 6 & 6 & 7 & 7 & 8 & 1 & 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 1 & 3 & 4 & 6 & 2 & 1 & 2 \\ 4 & 6 & 2 & 2 & 3 & 4 & 5 & 3 & 1 & 3 \\ 4 & 5 & 3 & 3 & 4 & 5 & 6 & 4 & 2 & 3 \\ 5 & 6 & 4 & 4 & 5 & 6 & 7 & 6 & 3 & 4 \\ 6 & 7 & 6 & 6 & 6 & 7 & 7 & 8 & 4 & 5 \end{bmatrix}$$

dobija se Moore-Penroseov inverz

$$A^\dagger = \left[\begin{matrix} 0.294371 & -0.168563 & -1.51113 & 1.41468 & -0.390817 \\ -0.0665814 & -0.0736365 & 1.22658 & -1.06364 & 0.229515 \\ 0.226663 & -0.178769 & -0.692492 & 0.705403 & -0.213752 \\ -0.164739 & 0.547244 & -0.656536 & 0.375969 & -0.115828 \\ -0.296725 & -0.0104986 & 1.03479 & -0.972437 & 0.359073 \\ -0.00818833 & -0.473753 & 1.66304 & -1.31891 & 0.352318 \\ 0.0619243 & 0.368475 & -1.34903 & 1.08137 & -0.32958 \\ -0.0806039 & 0.157918 & 0.0289882 & -0.143698 & -0.0341059 \\ 0.195087 & -0.484252 & 1.19782 & -0.791344 & 0.211391 \\ -0.0806039 & 0.157918 & -0.971012 & 0.856302 & -0.0341059 \end{matrix} \right.$$

$$
\left.\begin{matrix}
0.0414486 & -0.0402081 & -0.259697 & 0.454236 & -0.264066 & 0.230377 \\
0.000526393 & -0.192068 & 0.649101 & -0.103239 & -0.190912 & -0.102084 \\
-0.00878296 & -0.253651 & -0.237774 & 1.51422 & -1.31922 & 0.448655 \\
0.0147536 & 0.179273 & -0.196037 & -0.456054 & 0.531094 & -0.104233 \\
0.107833 & 0.237939 & 0.0435639 & -1.06532 & 0.937535 & -0.365726 \\
-0.102915 & -0.428182 & 0.224424 & 1.82997 & -1.84384 & 0.414315 \\
0.00597066 & 0.425623 & -0.433811 & -0.441829 & 0.711873 & -0.155578 \\
-0.0212507 & 0.0871716 & -0.0192523 & -1.49888 & 1.44795 & -0.138105 \\
0.00491787 & -0.190242 & 0.267988 & 0.764649 & -0.906302 & 0.048589 \\
-0.0212507 & 0.0871716 & -0.0192523 & -0.498879 & 0.447946 & -0.138105
\end{matrix}\right] .
$$

Primer 3.6 Na sledećoj slici je dat izgled ekrana prilikom izračunavanja generalisanog inverza za matricu $A_7 \times 6$.



Slika 1.

3.2 METODA PREGRADJIVANJA

Grevile [33] rekurzivni algoritam za izračunavanje generalisanih inverza može se upotrebiti i u programskim paketima koji koriste baze podataka. Algoritam je opisan u prethodnim glavama, pa je u daljem tekstu data samo njegova implementacija pomoću baze podataka.

Definišimo matrični tip podataka na sledeći način:

```
type  Matrica = record
                 m,n:integer;
                 Element:array[1..80,1..80] of real;
                end;
      NizMatrica=array[1..10] of Matrica;
public A,R,Z:NizMatrica;
```

Pomoćne funkcije za realizaciju algoritma su:

- funkcija $isZero(\,)$ sa formalnim parametrom A tipa $Matrica$ koja vraća True ili False u zavisnosti dali je matrica A nula matrica ili ne;

```
function TForm1.isZero(A:Matrica):boolean;
var i,j:integer; Nula:boolean;
begin
 Nula:=True;
 for i:=1 to A.m do
     for j:=1 to A.n do
       if A.Element[i,j]<>0 then Nula:=False;
 isZero:=Nula;
end;
```

- funkcija $MatNum(\,)$ koja za matricu $1x1$ vraća vrednost njenog jedinog elementa;

```
function TForm1.MatNum(A:Matrica):Real;
begin
 if (A.m=1) and (A.n=1) then MatNum:=A.Element[1,1];
end;
```

- funkcija $ithCol(\,)$ vraća i-tu kolonu matrice;

```
function TForm1.ithCol(i:integer; A:Matrica):Matrica;
var R:Matrica;k:integer;
begin
 R.n:=1; R.m:=A.m;
 for k:=1 to A.m do R.Element[k,1]:=A.Element[k,i];
 ithCol:=R;
end;
```

- funkcija $FirstiCol()$ izdvaja prvih i kolona matrice;

```
function TForm1.FirstiCol(i:integer; A:Matrica):Matrica;
var R:Matrica;b,k:integer;
begin
 R.n:=i; R.m:=A.m;
 for b:=1 to i do
   for k:=1 to A.m do R.Element[k,b]:=A.Element[k,b];
 FirstiCol:=R;
end;
```

- funkcija $AppRow(X,Y:Matrica)$ matrici X dodaje matricu Y kao m+1 vrstu matrica Y je matrica vrsta;

```
function TForm1.AppRow(X,Y:Matrica):Matrica;
var i,j:integer;
begin
 for i:=1 to Y.m do
     for j:=1 to X.n do X.Element[X.m+i,j]:=Y.Element[i,j];
 X.m:=X.m+Y.m;
 AppRow:=X;
end;
```

- funkcija $ZeroMat()$ vraća nula matricu mxn;

```
function TForm1.ZeroMat(m,n:integer):Matrica;
var i,j:integer; R:Matrica;
begin
 R.m:=m; R.n:=n;
 for i:=1 to m do
     for j:=1 to n do R.Element[i,j]:=0;
 ZeroMat:=R;
end;
```

- funkcija $SubsMat(X,Y:Matrica)$ oduzima maticu Y od matrice X, uz pretpostavku da su istih dimenzija;

```
function TForm1.SubsMat(X,Y:Matrica):Matrica;
var i,j:integer;
begin
  for i:=1 to X.m do
  for j:=1 to X.n do
    if abs(X.Element[i,j]-Y.Element[i,j])<1.0e-10
       then X.Element[i,j]:=0
       else X.Element[i,j]:=X.Element[i,j]-Y.Element[i,j];
 SubsMat:=X;
end;
```

- funkcija $MultSk(k:Real;A:Matrica)$ množi matricu A skalarem k;

```
function TForm1.MultSk(k:Real;A:Matrica):Matrica;
var i,j:integer;
begin
 for i:=1 to A.m do
  for j:=1 to A.n do A.Element[i,j]:=k*A.Element[i,j];
 MultSk:=A;
end;
```

- funkcija $Trans(A : Matrica)$ daje transponovanu matricu matrice A;

```
function TForm1.Trans(A:Matrica):Matrica;
var i,j:integer; T:Matrica;
begin
 for i:=1 to A.m do
  for j:=1 to A.n do T.Element[j,i]:=A.Element[i,j];
 T.n:=A.m; T.m:=A.n; Trans:=T;
end;
```

- funkcija $MultMat(X : Matrica; Y : Matrica)$ daje proizvod matrica X i Y redom;

```
function TForm1.MultMat(X:Matrica; Y:Matrica):Matrica;
var R:Matrica; i,j,k:integer;
begin
 R.m:=0; R.n:=0;
 if X.n=Y.m then
 with R do
 begin
     R:=ZeroMat(X.m,Y.n);
     for i:=1 to m do
     for j:=1 to n do
     for k:=1 to X.n do
         Element[i,j]:=Element[i,j]+X.Element[i,k]*Y.Element[k,j];
 end;
 MultMat:=R;
end;
```

Procedura $ReadFile(varA : Matrica; ime : String)$ učitava elemente matrice iz fajla.

```
procedure TForm1.ReadFile(var A:Matrica;ime:String);
var fajl:TextFile;i,j:integer;
begin
 AssignFile(fajl,ime); Reset(fajl);
 Read(fajl,A.m); ReadLn(fajl,A.n);
 for i:=1 to A.m do
  begin for j:=1 to A.n do  Read(fajl,A.Element[i,j]);
       ReadLn(fajl);
  end;
```

```
 CloseFile(fajl);
end;

function TForm1.Grevile(A:Matrica):Matrica;
var P,AR,AA,BB,CC,DD:Matrica; pom:Real;i:integer;
begin
     {Korak 1}
 AA:=ithCol(1,A);
 if isZero(AA) then AR:=Trans(AA)
              else begin
                   AR:=MultMat(Trans(AA),AA); pom:=MatNum(AR);
                   AR:=MultSk(1/pom,Trans(AA));
              end;
     {Korak 2}
 for i:=2 to A.n do
 begin
     {Korak 2.1}
 DD:=MultMat(AR,ithCol(i,A)); WriteTable(i+1,DD);
      {Korak 2.2}
 CC:=SubsMat(ithCol(i,A),MultMat(FirstiCol(i-1,A),DD));
      {Korak 2.3}
 if isZero(CC) then
  begin
     P:=MultMat(Trans(DD),DD); pom:=1+MatNum(P);
     BB:=MultSk(1/pom,MultMat(Trans(AR),DD));
  end else begin
           P:=MultMat(Trans(CC),CC); pom:=MatNum(P);
           BB:=MultSk(1/pom,CC);
           end;
      {Korak 2}
 AR:=SubsMat(AR,MultMat(DD,Trans(BB)));
 AR:=AppRow(AR,Trans(BB));
 end;
 Grevile:=AR;
end;
```

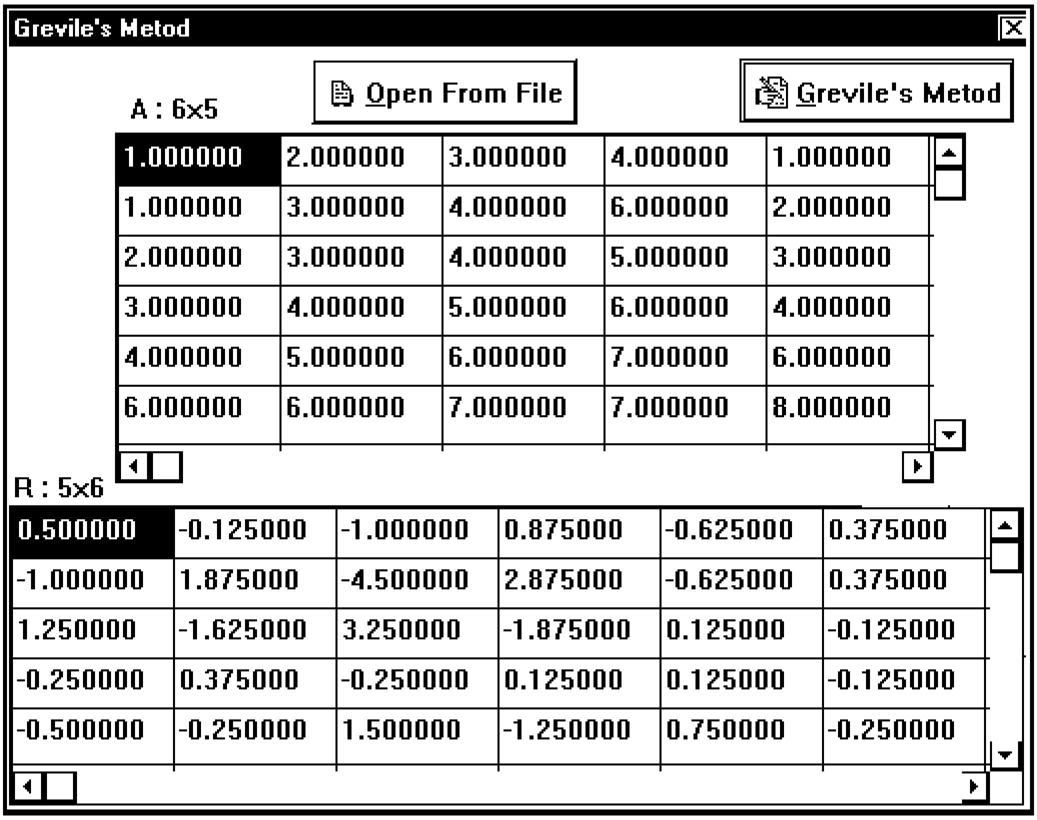
Primer 3.7 Za matricu A_{10*11} iz Primera 3.5 primenom funkcije $Grevile(A)$ dobija se Moore-Penrose inverz:

$$
\begin{array}{rrrrrl}
0.294371 & -0.168563 & -1.51113 & 1.41468 & -0.390817 & \dots \\
-0.0665814 & -0.0736365 & 1.22658 & -1.06364 & 0.229515 & \dots \\
0.226663 & -0.178769 & -0.692492 & 0.705403 & -0.213752 & \dots \\
-0.164739 & 0.547244 & -0.656536 & 0.375969 & -0.115828 & \dots \\
-0.296725 & -0.0104986 & 1.03479 & -0.972437 & 0.359073 & \dots \\
-0.00818833 & -0.473753 & 1.66304 & -1.31891 & 0.352318 & \dots \\
0.0619243 & 0.368475 & -1.34903 & 1.08137 & -0.32958 & \dots \\
-0.0806039 & 0.157918 & 0.0289882 & -0.143698 & -0.0341059 & \dots \\
0.195087 & -0.484252 & 1.19782 & -0.791344 & 0.211391 & \dots \\
-0.0806039 & 0.157918 & -0.971012 & 0.856302 & -0.0341059 & \dots
\end{array}
$$

...	0.0414486	−0.0402081	−0.259697	0.454236	−0.264066	0.230377
...	0.000526393	−0.192068	0.649101	−0.103239	−0.190912	−0.102084
...	−0.00878296	−0.253651	−0.237774	1.51422	−1.31922	0.448655
...	0.0147536	0.179273	−0.196037	−0.456054	0.531094	−0.104233
...	0.107833	0.237939	0.0435639	−1.06532	0.937535	−0.365726
...	−0.102915	−0.428182	0.224424	1.82997	−1.84384	0.414315
...	0.00597066	0.425623	−0.433811	−0.441829	0.711873	−0.155578
...	−0.0212507	0.0871716	−0.0192523	−1.49888	1.44795	−0.138105
...	0.00491787	−0.190242	0.267988	0.764649	−0.906302	0.048589
...	−0.0212507	0.0871716	−0.0192523	−0.498879	0.447946	−0.138105

Primer 3.8 Na sledećoj slici je dat izgled ekrana prilikom izračunavanja generalisanog inverza za matricu

$$A_{6\times 5} = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 \\ 1 & 3 & 4 & 6 & 2 \\ 2 & 3 & 4 & 5 & 3 \\ 3 & 4 & 5 & 6 & 4 \\ 4 & 5 & 6 & 7 & 6 \\ 6 & 6 & 7 & 7 & 8 \end{bmatrix}.$$



Slika 2.

3.3 METODA INVERZIJE INTERPOLACIJOM

Za implementaciju algoritma interpolacije pogodno je da vrednost interpolacionog polinoma prolazi kroz mali broj tačaka. Ovde su korišćene dve tačke s_0 i s_1 za implementaciju u paketu DELPHI. U tom slučaju je

$$\begin{aligned} L_0(s) &= \frac{s-s_1}{s_0-s_1} = \frac{1}{s_0-s_1}s + \frac{s_1}{s_1-s_0} = l_{0,1}s + l_{0,0} \qquad (3.3.1)\\ L_1(s) &= \frac{s-s_0}{s_1-s_0} = \frac{1}{s_1-s_0}s + \frac{s_0}{s_0-s_1} = l_{1,1}s + l_{1,0}\\ R_0 &= A_0^\dagger l_{0,0} + A_1^\dagger l_{1,0} = A_0^\dagger \frac{s_1}{s_1-s_0} + A_1^\dagger \frac{s_0}{s_0-s_1} \qquad (3.3.2)\\ R_1 &= A_0^\dagger l_{0,1} + A_1^\dagger l_{1,1} = A_0^\dagger \frac{1}{s_0-s_1} + A_1^\dagger \frac{1}{s_1-s_0} \end{aligned}$$

Inverzija polinomilalnih matrica interpolacijom koristi modifikovani *Algoritam 2.10* u *Koraku 4.* sa dve početne tačke s_0 i s_1, u formi (3.3.1) i (3.3.2) data je sa:

Veći broj funkcija je opisan ranije. Za implementaciju je potrebno modifikovati samo deo za čitanje i upis elemenata elemenata polinomijalne matrice iz fajla i u fajl na sledeći način:

```
procedure TForm1.ReadFile(var A:NizMatrica;ime:String);
var fajl:TextFile;  k,i,j,m,n,dim:integer;
begin
     AssignFile(fajl,ime);Reset(fajl);
     Read(fajl,dim);Read(fajl,m);ReadLn(fajl,n);
     for k:=1 to dim do
     begin
        A[k].m:=m; A[k].n:=n;
        for i:=1 to m do
        begin
          for j:=1 to n do Read(fajl,A[k].Element[i,j]);
          ReadLn(fajl);
        end;
        ReadLn(fajl);
     end;
     CloseFile(fajl);
end;

procedure TForm1.WriteR(A:NizMatrica);
var i,j:integer;
begin
 for i:=0 to 80 do
  for j:=0 to 80 do
 begin
```

```
 StringGridR1.Cells[i,j]:=''; StringGridR2.Cells[i,j]:='';
         end;
 Label3.Caption:='R[1] : '+IntToStr(A[1].m)+'x'+IntToStr(A[1].n);
     for i:=1 to A[1].m do
         for j:=1 to A[1].n do
 StringGridR1.Cells[j-1,i-1]:=FloatToStrF(A[1].Element[i,j], ffFixed, 6, 2);
 Label4.Caption:='R[2] : '+IntToStr(A[2].m)+'x'+IntToStr(A[2].n);
     for i:=1 to A[2].m do
      for j:=1 to A[2].n do
 StringGridR2.Cells[j-1,i-1]:=FloatToStrF(A[2].Element[i,j], ffFixed, 6, 2);
end;
```

Procedura $WriteFile(\,)$ upisuje rezultujuće matrice u fajl "*rez.txt*".

```
procedure TForm1.WriteFile(A:NizMatrica);
var fajl:TextFile;
    i,j,k:integer; ime1:string;
begin
     ime1:='rez.txt'; AssignFile(fajl,ime1); Rewrite(fajl);
     for k:=1 to dim do
     begin
      for i:=1 to A[k].m do
      begin
         for j:=1 to A[k].n do Write(fajl,A[k].Element[i,j]:6:2,' ');
         WriteLn(fajl);
      end;
      WriteLn(fajl);
     end;
     CloseFile(fajl);
end;
```

Interpolacija je realizovana sa:

```
procedure TForm1.InterpolClick(Sender: TObject);
var t1,t2:real;
begin
  {t1 is first interpolation point}
    t1:=strtofloat(first.Text);
  {t2 is second interpolation point}
    t2:=strtofloat(seconds.Text);
    Z[1]:=Grevile(SubsMat(MultSk(t1,A[1]),MultSk(-1,A[2])));
    Z[2]:=Grevile(SubsMat(MultSk(t2,A[1]),MultSk(-1,A[2])));
    R[1]:=SubsMat(MultSk(1/(t1-t2),Z[1]),MultSk(1/(t1-t2),Z[2]));
    R[2]:=SubsMat(MultSk(t2/(t2-t1),Z[1]),MultSk(t1/(t2-t1),Z[2]));
  {Write results matrices on screen}
    WriteR(R);
  {Write results matriced in file}
    WriteFile(R);
end;
```

Primer 3.9 Startne tačke 1 i 2 su u *Edit* komponentama 'first.Text' i 'seconds.Text' respektivno, a matrica S_{11} je iz [86]:

$$S11 = \begin{pmatrix} a+1 & a & a & a & a & a & a & a & a & a & a+1 \\ a & a-1 & a & a & a & a & a & a & a & a & a \\ a & a & a+1 & a & a & a & a & a & a & a & a \\ a & a & a & a-1 & a & a & a & a & a & a & a \\ a & a & a & a & a+1 & a & a & a & a & a & a \\ a & a & a & a & a & a-1 & a & a & a & a & a \\ a & a & a & a & a & a & a+1 & a & a & a & a \\ a & a & a & a & a & a & a & a-1 & a & a & a \\ a & a & a & a & a & a & a & a & a+1 & a & a \\ a & a & a & a & a & a & a & a & a & a-1 & a \\ a+1 & a & a & a & a & a & a & a & a & a & a+1 \end{pmatrix}.$$

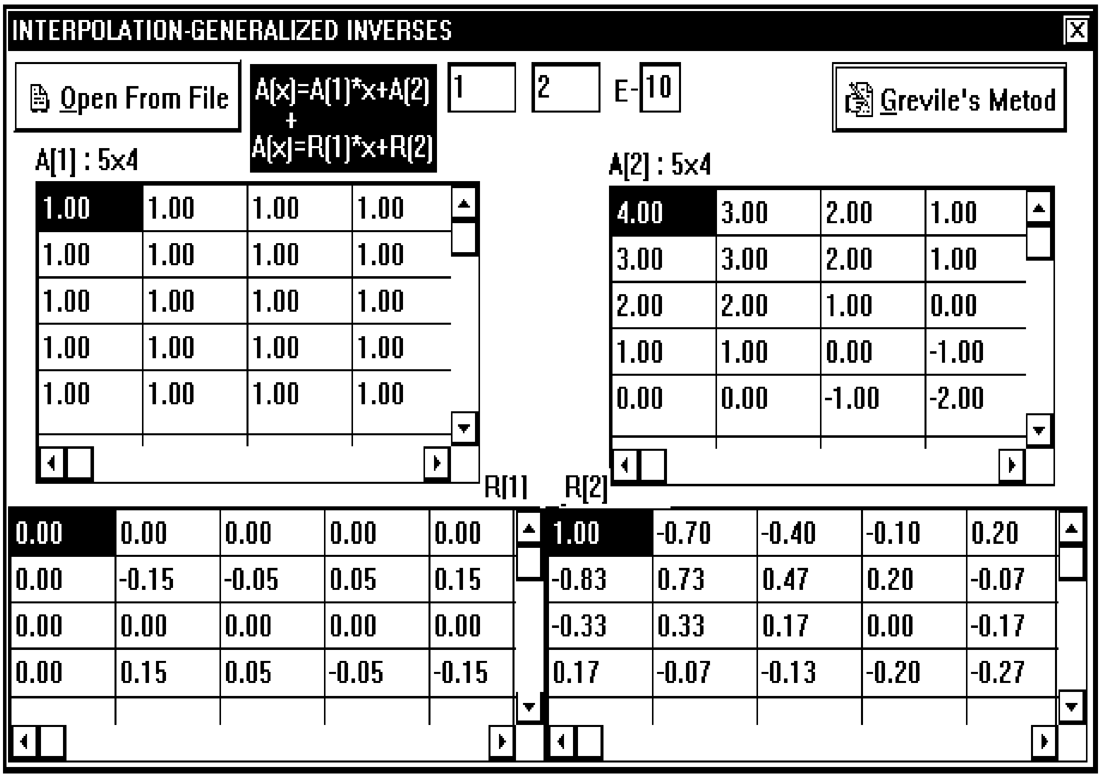
Ulazni fajl je u formi:

```
2 11 11
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00

1.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  1.00
0.00 -1.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  0.00  1.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  0.00  0.00 -1.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  0.00  0.00  0.00  1.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  0.00  0.00  0.00  0.00 -1.00  0.00  0.00  0.00  0.00  0.00
0.00  0.00  0.00  0.00  0.00  0.00  1.00  0.00  0.00  0.00  0.00
0.00  0.00  0.00  0.00  0.00  0.00  0.00 -1.00  0.00  0.00  0.00
0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  1.00  0.00  0.00
0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00 -1.00  0.00
1.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  1.00
```

Rezultujuća matrica nalazi se u tekstualnom fajlu 'rez.txt' u formi:

```
-0.25  0.50 -0.50  0.50 -0.50  0.50 -0.50  0.50 -0.50  0.50 -0.25
 0.50 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  0.50
-0.50  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -0.50
 0.50 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  0.50
-0.50  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -0.50
 0.50 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  0.50
-0.50  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -0.50
 0.50 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  0.50
-0.50  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -0.50
 0.50 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  1.00 -1.00  0.50
-0.25  0.50 -0.50  0.50 -0.50  0.50 -0.50  0.50 -0.50  0.50 -0.25

 0.25  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.25
 0.00 -1.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
 0.00  0.00  1.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00 -1.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  1.00  0.00  0.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00 -1.00  0.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  1.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  0.00 -1.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  1.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00 -1.00  0.00
 0.25  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.25
```

Primer 3.10 Na sledećoj slici je dat izgled ekrana prilikom implementacije.



Slika 3.

Primer 3.11 Efektivnost *Algoritma 2.10* u paketima MATHEMATICA 4.0 i DELPHI 6.0 u odnosu na *CPU* vreme, na Pentium II procesoru na 433 MHz. Sa S_n, A_n i F_n su označene test matrice iz [86], u tački $\{1,2\}$.

Dimension	Test matrix	MATHEMATICA	DELPHI
4	S	0.22 Second	0.51 Second
4	A	0.22 Second	0.51 Second
4	F	0.22 Second	0.51 Second
11	S	0.27 Second	1.01 Second
11	A	0.27 Second	1.01 Second
11	F	0.27 Second	1.01 Second
35	S	4.50 Second	2.01 Second
35	A	2.52 Second	2.01 Second
35	F	2.47 Second	2.01 Second
50	S	12.64 Second	3.01 Second
50	A	6.09 Second	3.01 Second
50	F	5.54 Second	3.01 Second
79	S	48.17 Second	5.01 Second
79	A	17.52 Second	5.01 Second
79	F	16.31 Second	5.01 Second

Tabela 2.

4 ZAKLJUČAK

Moj cilj nije bio da ispitujem implementaciju svih poznatih metoda za izračunavanje generalisanih metoda. Takođe mi nije bio cilj da poredim direktne i iterativne metode. Ja sam proučavao pojavu problema povećanje broja operacija u pokretnom zarezu, tokom iteracije, prouzrokovane ponovnim izračunaranjem pojedinih izraza, u numeričkim izračunavanjima pseudoinverza, kao i vezu proceduralnih i neproceduralnih programskih jezika i njihovu mogućnost u uzračunavanju generalisanih inverza.

Predložio sam nove metode i nove algoritme, kao i modifikacije postojećih metoda i algoritama, koji rešavaju pomenute probleme. Sve je bazirano na mogućnostima paketa MATHEMATICA kao tipičnog primera programskog paketa za simboličko izračunavanje, i paketa DELPHI kao predstavnika proceduralnih programskih jezika.
Metodi i algoritmi koje sam predložio implementirani su i testirani na numeričkim i racionalnim matricama, polinomijalnim matricama, retkoposednutim matricama i na bazama podataka. Oni su primenljivi na sve proceduralne programske jezike.

Pokazao sam da je u nekim situacijama pogodnije koristiti proceduralne programske jezike za izračunavanje generalisanih inverza. Kroz veliki broj primera testirao sam algoritme za izračunavanje generalisanih inverza, i vršio upoređenje pojedinih metoda u paketu MATHEMATICA i DELPHI.

Uspostavio sam relaciju između baza podataka i generalisanih inverza i preporučio ovo rešenje kao koristan alat u sličnim problemima i situacijama.

LITERATURA

[1] Aleksić, V. , Rakočević, V. *Approximate properties of the Moore-Penrose inverse*, VIII Conference on Applied Mathematics, Tivat (1993), 1–14

[2] S. Barnett *Leverrier's algorithm: a new proof and extensions*, SIAM J. Matrix Anal. Appl. **10** (1989), 551–556.

[3] Bapat,R.B. , K.P.S.B. Rao and K.M. Prasad *Generalized inverses over integral domains* Linear Algebra Appl. **140** (1990), 181–196

[4] Bapat, R.B. *Generalized inverses with proportional minors* Linear Algebra Appl. **211** (1994), 27–35

[5] Ben-Israel, A. and Grevile, T.N.E. *Generalized inverses: Theory and applications* Wiley-Interscience, New York, (1974)

[6] Ben-Israel, A. *On matrices of index zero or one* SIAM J. Appl. Math. **17** (1969), 1118–1121

[7] Ben-Israel, A. *A volume associated with $m \times n$ matrices* Linear Algebra Appl. **167** (1992), 87–111

[8] Bischof, C.H. , Bouaricha,A., Khademi, P.M. and Moré, J.J. *Computing gradients in large-scale optimization using automatic differentiation* Technical Report, (1995)

[9] Campbell, S.L. , Meyer, C.D. *Generalized inverses of Linear Transformations* Pitman, New York, (1979)

[10] B. Carnahan, H.A. Luther, and J.O.Wilkes, *Applied Numerical Methods* New York: Wiley,(1969)

[11] Calvert, C. *Delphi Unleashed* Sams Publishing, (1995)

[12] Calvert, C. DELPHI *punom snagom* CET Computer Equipment and Trade, Beograd, (1996)

[13] Davenport,J.H., Siret, Y. and Tournier, E. *Computer Algebra*, Academic Press, (1988).

[14] H.P. Decell *An application of the Cayley-Hamilton theorem to generalized matrix inversion*, SIAM Review **7** No 4 (1965), 526–528.

[15] Dennis, J.E., Moré, J.J. *Quasi-Newton methods, motivation and theory* SIAM Rev. **19** No 1.(1977), 46–89

[16] D.K. Faddeev and V.N. Faddeeva *Computational Methods of Linear Algebra*, Freeman, San Francisko, 1963.

[17] J.S. Frame *A simple recursion formula for inverting a matrix*, Bull. Amer. Math. Soc. **55** (1949), 19–45.

[18] Fragulis, G.,Mertzios, B.G. and Vardulakis, A.I.G. *Computation of the inverse of a polynomial matrix and evaluation of its Laurent expansion*,Int. J. Control, **53** (1991),431–443

[19] J. Ji, *An alternative limit expression of Drazin inverse and its applications*, Appl. Math. Comput. **61** (1994), 151–156.

[20] J. Ji, *A finite algorithm for the Drazin inverse of a polynomial matrix*, Appl. Math. Comput., **30** (2002), 243–251.

[21] Ben-Israel, A. *On matrices of index zero or one* SIAM J. Appl. Math. **17** (1969), 1118–1121

[22] Jones, J., Karampetakis, N.P. and Pugh, A.C. *The computation and application of the generalized inverse vai Maple* J. Symbolic Computation,**25** (1998),99–124

[23] Karampetakis, N.P. *Computation of the generalized inverse of a polynomial matrix and applications*, Linear Algebra Appl.,**252** (1997), 35–60

[24] Karampetakis, N.P. *Generalized inverses of two-variable polynomial matrices and applications*, Circuits Systems Signal Processing,**16** (1997),439–453

[25] Karampetakis, N.P. and Tzekis, P. *On the computation of the generalized inverse of a polynomial matrix* 6th Medit. Symposium on New Directions in Control and Automation,(1998), 1–6

[26] Karlin S. *Mathematical Methods and Theory in Games* Programming and Economics, Pergman Press, London (1959)

[27] Kraus L. *Programsko okruženje* DELPHI *sa rešenim zadacima*, Mikro knjiga, Beograd, (1996)

[28] Kovačević V.V.-Vujčić *A view of interior point methods for linear programming* YUJOR **5** (1995), 173–193

[29] Kuhin, H.W. and Tucker, A.W. *Nonlinear programming* Proceedings of the second Berkeleg Symposium on Mathematical Statistics an Probability, California (1951)

[30] Lovass Nagy, V., Miller, R. and Powers, D. *Transfer function matrix synthesis by matrix generalized inverses* Int. J. Control, **27** (1978), 387–391

[31] Lovass Nagy, V., Miller, R. and Powers, D. *Further results on output control in the servomechanism sence* Int. J. Control,**27** (1978),133–138

[32] Lovass Nagy, V., Miller, R. and Powers, D. *An introduction to the application of the simplest matrix-generalized inverse in system science*, IEEE Trans. Auto. Control, **25** (1978), 766–771

[33] Grevile, T.N.E. *Some applications of the pseudo-inverse of matrix* SIAM Rev. **3** (1960), 15–22

[34] Golub, G.H. and Van Loan, C.F. *Matrix Computations* The Johns Hopkins University Press, Baltimore Maryland, (1984)

[35] Groetch, C.W. *Generalized Inverses of Linear Operators* Marcel Dekker, Inc. New York and Basel, (1977)

[36] R.E. Hartwig *More on the Souriau-Frame algorithm and the Drazin inverse*, SIAM J. Appl. Math. **31** No 1 (1976), 42–46.

[37] Henderson, K. DELPHI 3 *Client-Server Developer's Guide* Borland PRESS, (1997)

[38] Miao, J. *Reflexive generalized inverses and their minors* Linear and Multilinear Algebra **35** (1993), 153–163

[39] Milovanović, G.V. *Numerička Analiza I* Naučna Knjiga, Beograd, (1991)

[40] Milovanović, G.V. *Numerička Analiza II* Naučna Knjiga, Beograd, (1991)

[41] Milovanović, G.V., Mitrinović, D.S., Rassias,Th.M. *Topics in Polynomials: Extremal Problems, Inequalities*, Zeros World Scientific, Singapore-New Jersey-London-Hong Kong, (1994)

[42] Milovanović, G.V., Stanimirović, P.S. *Block reprezentation of the group inverse* Proceedings of the II Mathematical Conference in Priština (1996), 79–88

[43] Milovanović, G.V., Stanimirović, P.S. *Rank factorization and Moore-Penrose inverse* XI Conference on Applied Mathematics (1997), 235–250

[44] Milovanović, G.V., Stanimirović, P.S. *On Moore-Penrose inverse of block matrices and full-rank factorization* Publ.Inst.Math., **76**(1997), 26–40

[45] Mitra,S.K., Rao R.C. *Extensions of a duality theorem concerning g-inverses of matrices* The Indian Journal of Statistics, **37** (1975), 439–445

[46] Noble, B. *Methods for computing the Moore-Penrose generalized inverse, and related matters* Generalized Inverses and Applications edited by M.Z. Nashed, Academic Press, New York (1976), 245–301

[47] Penrose, R. *A generalized inverse for matrices* Proc. Cambridge Philos. Soc.,**51** (1955), 406–413

[48] Penrose, R. *On best approximate solution of linear matrix equations* Proc. Cambridge Philos. Soc.,**52** (1956), 17–19

[49] Petrić, J. *Operaciona Istraživanja*, Naučna Knjiga, Beograd, (1989)

[50] Prasad K.M. , Rao K.P.S.B. and Bapat, R.B. *Generalized inverses over integral domains. II. Group inverses and Drazin inverses* Linear Algebra Appl. **146** (1991), 31–47

[51] Prasad, K.M. and Bapat, R.B. *The Generalized Moore-Penrose inverse* Linear Algebra Appl. **165** (1992), 59–69

[52] Pyle, L.D. *The weighted generalized inverse in nonlinear programming-active set selection using a variable-metric generalization of the simplex algorithm* International symposium on extremal methods and systems analysis, Lecture Notes in Economics and Mathematical Systems,**174** (1977), 197–231

[53] Pyle, L.D. *The generalized inverse linear programming. Basic structure* SIAM J. Appl. Math. **22** No 3 (1972), 335–355

[54] Rao,M. K. Subramanian, K. Krishnamurty, E. *Residue arithmetic algorithms for exact computation of g-inverse of matrices* SIAM J. Numer. Anal. **13** (1976), 155–171

[55] Rao,C.R. and Mitra,S.K. *Generalized Inverse of Matrices and its Applications* John Wiley and Sons, Inc, New York, London, Sydney, Toronto, (1971)

[56] Robinson, D.W. *The image of the adjoint mapping* Linear Algebra Appl. **277** (1998), 143–148

[57] Rosen,J.B. *Minimum and basic solutions to singular linear systems* SIAM **12** (1964), 156–162

[58] Shinozaki, N. , Sibuya, M. and Tanabe, K. *Numerical algorithms for the Moore-Penrose inverse of a matrix: direct methods* Annals of the Institute of Statistical Mathematics **24**, No. 1 (1972), 193–203

[59] Shinozaki, N. , Sibuya, M. and Tanabe, K. *Numerical algorithms for the Moore-Penrose inverse of a matrix: iterative methods* Annals of the Institute of Statistical Mathematics **24**, No. 1 (1972), 621–629

[60] Stanimirović, P.S. *General determinantal representation of generalized inverses over integral domains* Publicationes Mathematicae Debrecen **54** (1999), 221–249

[61] Stanimirović, P.S. and Đorđević, D.S. *Full-rank and determinantal representation of the Drazin inverse* Linear Algebra and its Applications **311** (2000), 31–51

[62] Stanimirović, P.S. *Limit representations of generalized inverses and related methods* Appl. Math. Comput. **103** (1999), 51–68

[63] Stanimirović, P.S. and Stanković, M. *Generalized algebraic complement and Moore- Penrose inverse*, Filomat **8** (1994), 57-64.

[64] Stanimirović, P.S. *Computing pseudoinverses using minors of an arbitrary matrix* Filomat **9:2** (1995), 285–294

[65] Stanimirović, P.S., Tasić, M.B. and Rančić, S. *Repetitive applications of functions as arguments in programming languages* Proceedings of VIII Conference on Logic and Computer Science, LIRA '97, Novi Sad (1997), 231–238

[66] Stanimirović *Computing minimum and basic solutions of linear systems using the Hyper-power method* Studia Scientiarum Mathematicarum Hungarica, **35** (1999), 175–184

[67] Stanimirović, P.S. *General determinantal reprezentation of pseudoinverses and its computation* Rev. Academia de Ciencias Zaragoza,**50** (1995), 41–50

[68] Stanimirović, P.S. *Computation of determinants and inverses of rectangular or singular matrices using residug aritmetic* Publ. Inst. Math.**60** (1996), 15–26

[69] Stanimirović, P.S. *Block reprezentation of* $\{2\}$*,* $\{1{,}2\}$ *inverses and the Drazin inverse of matrices* Indian J. Pure. Apl. Math., **29** (1998), 1159–1176

[70] Stanimirović, P.S. *Implementation of block representation of the Moore-Penrose inverse* Proceedings of the II Mathematical Conference in Priština (1996), 205–214

[71] Stanimirović, P.S. *Limit representations of generalized inverses and related methods* Appl. Math. Comput.,**103** (1999), 54–68

[72] Stanimirović, P.S. *Determinantal representation of* $\{i, j, k\}$ *inverses and solution of linear systems* Mathematica Slovaca **49** (1999), Accepted

[73] Stanimirović, P.S. *A finite algorithm for generalized inverses of polynomial and rational matrices* Appl. Math. Comput., Accepted

[74] Stanimirović, P.S., Tasić, M.B. *A problem in computation of pseudoinverses*, Appl. Math. Comput., **135** (2-3) (2003), 443–469

[75] Stanimirović, P.S., Tasić, M.B. *Drazin Inverse of One-Variable Polynomial Matrices*, Filomat, Accepted

[76] Stanimirović P.S., Tasić, M.B., Ristić, M., *Simbolic implementation of the Hooke-Jeeves method*, YUJOR **Vol 9** (1999) No. 2, 285–300.

[77] Stanimirović P.S., Tasić, M.B. *Computing determinal representation of generalized inverses*, Korean J. Comput & Appl. Math. **Vol 9** (2002) No. 2, 349–359.

[78] Stanimirović P.S. and Karampetakis N.P. *Symbolic implementation of Leverrier-Faddeev algorithm and applications*, 8th IEEE Medit. Conference on Control and Automation, Patra, Greece (2000).

[79] Stanimirović P.S., Karampetakis N.P. *On the computation of Drazin inverse of a rational matrix and applications*, Technical Report, Department of Mathematics, Aristotle University of Thessaloniki, Thessaloniki 54006, Greece, 2000.

[80] Tasić, M.B. , *Implementacija nekih metoda za izračunavanje ekstremnih vrednosti i generalisanih inverza*, Magistarska teza, (2000), Prirodno-matematički fakultet, Univerzitet u Nišu.

[81] Tanabe K. *Conjugate-gradient method for computing the Moore-Penrose inverse and rank of a matrix* Journal of Optimization Theory and Applications, **22**, No. 1 (1977), 1–23

[82] G. Wang and Y. Lin *A new extension of the Leverrier's algorithm* Linear Algebra Appl. **180** (1993), 227–238.

[83] Wolfram, S. *Mathematica: a System for Doing Mathematics by Computer* Addison-Wesley Publishing Co, Redwood City, California, (1991)

[84] Wolfram, S. *Mathematica Book, Version 3.0* Wolfram Media and Cambridge University Press, (1996)

[85] Zielke, G. *A survey of generalized matrix inverse* Banach Center Publication **13** (1984), 499–526

[86] Zielke, G. *Report on test matrices for generalized inverses* Computing **36** (1986), 105–162

[87] Zielke, G. *Some remarks on matrix norms, condition numbers, and error estimates for linear equations* Linear Algebra Appl.,**110** (1988), 29–41

[88] Zielke, G. *Iterative refinement of generalized matrix inverses now practicable* A.C.M. SIGNUM Newsletter **13.4** (1978), 9–10

[89] Zlobec,S. Petrić, J. *Nelinearno Programiranje*, Naučna Knjiga, Beograd, (1989)

[90] Žukovski, E.L. and Lipcer, R.S. *On computation pseudoinverse matrices*, Ž Vicisl. Mat. i Mat. Fiz. **15** (1975), 489–492, In Russian

[91] Žukovski, E.L. and Lipcer, R.S. *On recurent computation of normal solutions of linear algebraic equations* Ž Vicisl. Mat. i Mat. Fiz. **12** (1972), 843–857, 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**:	Милан Б. Тасић
Ментор, **МН**:	Предраг С. Станимировић
Наслов рада, **НР**:	**ИЗРАЧУНАВАЊЕ ГЕНЕРАЛИСАНИХ ИНВЕРЗА**
Језик публикације, **ЈП**:	српски
Језик извода, **ЈИ**:	енглески
Земља публиковања, **ЗП**:	Југославија
Уже географско подручје, **УГП**:	Србија
Година, **ГО**:	2003.
Издавач, **ИЗ**:	ауторски репринт
Место и адреса, **МА**:	Ниш, Вишеградска 33
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)	3 пог.; 163 стр. ; 91 реф.; 2 таб.; 3. сл.; граф. прикази
Научна област, **НО**:	математика
Научна дисциплина, **НД**:	нумеричка анализа
Предметна одредница/Кључне речи, **ПО**:	симболичко израчунавање, примењена математика, генералисани инверзи
УДК	519.688 (043.3) 519.613 (043.3)
Чува се, **ЧУ**:	библиотека
Важна напомена, **ВН**:	
Извод, **ИЗ**:	Заједнички проблем настао при имплементацији различитих метода за нумеричко израчунавање генералисаних инверза је брзо увећање операција у покретном зарезу. Већина алгоритама за нумеричко израчунавање генералисаних инверза је лоше условљена. Ова чињеница представља оправдање за одговарајуће алгоритме који користе могућности симболичке обраде података у програмском пакету MATHEMATICA и DELPHI. Уведени алгоритми су верификовани на великом броју познатих тест примера. Алгоритми су погодни за примену у процедуралним програмским језицима, који немају могућност симболичког израчунавања.
Датум прихватања теме, **ДП**:	5.7.2002. године
Датум одбране, **ДО**:	

Чланови комисије, **КО**:	Председник:	
	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**:	Milan B. Tasić
Mentor, **MN**:	Predrag S. Stanimirović
Title, **TI**:	COMPUTATION OF GENERALIZED INVERSES
Language of text, **LT**:	Serbian
Language of abstract, **LA**:	English
Country of publication, **CP**:	Yugoslavia
Locality of publication, **LP**:	Serbia
Publication year, **PY**:	2003
Publisher, **PB**:	author's reprint
Publication place, **PP**:	Niš, Višegradska 33
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)	3 cap.; 163 pag. ; 91 ref.; 2 tab.; 3. pic.; graphic representations
Scientific field, **SF**:	Mathematics,
Scientific discipline, **SD**:	numerical analysis
Subject/Key words, **S/KW**:	Symbolic Computation, Applied Mathematics, generalized inverses
UC	519.688 (043.3) 519.613 (043.3)
Holding data, **HD**:	library
Note, **N**:	
Abstract, **AB**:	A common problem arising in the implementation of various methods for numerical computation of generalized inverses is a rapid magnification of floating point operations. There is proposed a few different algorithms in the programming packages MATHEMATICA and DELPHI, to avoid this common problem. The first approach uses a possibility of the package MATHEMATICA to define functions which remember all values that they find. The second approach is based on the construction of several databases by means of the language DELPHI. Effectiveness of these algorithms is investigated and illustrated by appropriate examples.
Accepted by the Scientific Board on, **ASB**:	5.7.2002
Defended on, **DE**:	
Defended Board, **DB**: President:	
Member:	
Member, Mentor:	