Methods for computing generalized inverses of matrices

Predrag S. Stanimirović
Marko D. Petković
Milan B. Tasić

Faculty of Science and Mathematics
University of Niš, Serbia
peckois@ptt.rs, dexterofnis@gmail.com, milan12t@ptt.rs

Theoretical Computer Science - From Foundation to Application

Niš, 2009

Outline of the talk

1. Generalized inverses of constant matrices
   - Introduction
   - Definitions and basic properties
2. Methods based on generalized Cholesky factorization
   - Introduction
   - Recursive Cholesky factorization
   - Computing the MP inverse in matrix multiplication complexity
3. Generalized inverses of rational and polynomial matrices
   - Introduction and previous work
   - Interpolation method for computing Drazin inverse
   - Interpolation methods for other generalized inverses
   - Greville partitioning method

Outline of the talk

Outline of the talk

Outline of the talk

Introduction

- The concept of generalized inverses was first introduced by I. Fredholm in 1903, where a generalized inverse of an integral operator was given and was called "pseudoinverse"
- Even before, in some considerations of C. F. Gauss it is implicitly considered an idea about generalized inverses (in least-square principle for non-consistent linear systems)
- The generalized inverse of matrices was first introduced by E. Moore in 1920.
- Lately in 1955, R. Penrose showed that Moore's inverse is unique matrix satisfying four equations.
- This result reviewed the study of generalized inverses.

Theorem (Moore 1920)

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

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

Moore-Penrose and $\{i, j, \dots, k\}$ inverses

- R. PENROSE, *A generalized inverse for matrices*, Proc. Cambridge Philos. Soc. **51** (1955), 406-413.

Theorem (Penrose 1955)

For any matrix $A \in \mathbb{R}^{m \times n}$, the following system of matrix equations

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

has unique solution $X \in \mathbb{R}^{n \times m}$. This solution is known as Moore-Penrose inverse (generalized inverse) of matrix A and is denoted by $A^\dagger$.

If A is square regular matrix, then its inverse matrix A^{-1} trivially satisfy system (1).

Definition

Let $A\{i, j, \dots, k\}$ be set of matrices satisfies equations $(i), (j), \dots, (k)$ among $(1), \dots, (4)$ from (1). Such matrices are called $\{i, j, \dots, k\}$ inverses and denoted by $A^{(i,j,\dots,k)}$. The set of all $A^{(i,j,\dots,k)}$ is denoted by $A\{i, j, \dots, k\}$.

Theorem (Penrose 1955)

Let $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^{m\times 1}$. Minimum-norm least-squares solution of the system $Ax = b$ is given by $x^* = A^\dagger b$. All other least-squares solutions are given by

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

Lemma

Let $A \in \mathbb{C}^{m\times n}$. Then

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

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

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

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

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

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

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

Drazin inverz

- M.P. DRAZIN, *Pseudo inverses in associative rings and semigroups*, Amer. Math. Monthly **65**, (1958), 506-514.

Theorem (Drazin 1958)

Let $A \in \mathbb{C}^{n\times n}$ be arbitrary matrix and let $k = \operatorname{ind}(A)$. Then the following solution of matrix equations

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

has unique solution. This solution is called Drazin inverse of matrix A and denoted by A^D.

Lemma

Let $A \in \mathbb{C}^{n\times n}$ and let $k = \operatorname{ind}(A)$. Then holds

(1) $(A^*)^D = (A^D)^*$, $(A^T)^D = (A^D)^T$, $(A^n)^D = (A^D)^n$ *for any* $n = 1, 2, \ldots$,

(2) $((A^D)^D)^D = A^D$, $(A^D)^D = A$ *if and only if* $k = 1$,

(3) $R(A^D) = R(A^l)$ *and* $N(A^D) = N(A^l)$ *for every* $l \geq k$,

(4) *If* λ *is an eigenvalue of A then* $\lambda^\dagger$ *is an eigenvalue of* A^D.

Methods for computing generalized inverses of matrices

(1) Iterative methods

(2) Direct methods

- **(2.1)** Methods based on the full-rank factorization.
- **(2.2)** Block representations.
- **(2.3)** Methods based on singular value decomposition (SVD).
- **(2.4)** Leverrier-Faddev method.
- **(2.5)** Partitioning method.
- **(2.6)** Methods based on generalized Cholesky factorization.
- **(2.7)** Limit and determinant representations.

Outline of the talk

Methods based on generalized Cholesky factorization

- M.D. Petković, P.S. Stanimirović, *Generalized inversion is not harder than matrix multiplication*, Journal of Computational and Applied Mathematics, **230:1** (2009), 270–282.

Methods for computing Moore-Penrose and other $\{i, j, \ldots, k\}$ inverses in matrix multiplication time complexity.

Better time complexity than all other methods, $\mathcal{O}(n^{2+\epsilon})$ where $0 < \epsilon < 1$.

Applicable both for symbolic and numerical computation.

Generalized Cholesky factorization

Theorem

Let A be a symmetric, possibly singular, positive semi-definite matrix of the order $n \times n$. Then there is an upper triangular matrix $U = [u_{ij}]$ such that $U^*U = A$ and $u_{ii} \geq 0$ for all $i = 1, \ldots, n$. If for an index i one has $u_{ii} = 0$, then $u_{ij} = 0$ for all $j = 1, \ldots, n$. Moreover, the matrix U with these properties is unique.

The factorization $A = U^*U$ is generalized Cholesky factorization of A.

Example

$$A = \begin{bmatrix} 429 & 429 & 109 & 109 \\ 429 & 429 & 109 & 109 \\ 109 & 109 & 30 & 30 \\ 109 & 109 & 30 & 30 \end{bmatrix}, \quad U = \begin{bmatrix} \sqrt{429} & \sqrt{429} & \frac{109}{\sqrt{429}} & \frac{109}{\sqrt{429}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{\frac{989}{429}} & \sqrt{\frac{989}{429}} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Generalized inverses and Cholesky factorization

- P. COURRIEU, *Fast Computation of Moore-Penrose Inverse Matrices*, Neural Information Processing - Letters and Reviews, **8:2** (2005) 25–29.

Theorem (Courrieu 2005)

Let $A \in \mathbb{C}^{m \times n}$ and $A^*A = U^*U$ be the generalized Cholesky factorization of the matrix A^*A. If the matrix L^* is obtained from U by dropping zero rows

$$A^\dagger = L(L^*L)^{-2}L^*A^*.$$

Generalization concerning $\{1,2,3\}$, $\{1,2,4\}$, $\{2,3\}_s$ i $\{2,4\}_s$ is given by Stanimirović and Tasić in:

- P.S. STANIMIROVIĆ, M.B. TASIĆ, *Computing generalized inverses using LU factorization of matrix product*, International Journal of Computer Mathematics, doi: 10.1080/00207160701582077.

Recursive Cholesky factorization (Regular case)

Consider the block representation of the matrices A i U:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^* & A_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ \mathbb{O} & U_{22} \end{bmatrix}, \quad U_{11}, A_{11} \in \mathbb{C}^{k \times k}.$$

Equation $A = U^*U$ is equivalent with the following system:

$$A_{11} = U_{11}^* U_{11}, \quad A_{12} = U_{11}^* U_{12}, \quad A_{22} = U_{12}^* U_{12} + U_{22}^* U_{22}.$$

Consider the same block decomposition of the matrix $Y = U^{-1}$. Then holds:

$$\begin{bmatrix} U_{11} & U_{12} \\ \mathbb{O} & U_{22} \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ \mathbb{O} & Y_{22} \end{bmatrix} = \begin{bmatrix} U_{11}Y_{11} & U_{11}Y_{12} + U_{12}Y_{22} \\ \mathbb{O} & U_{22}Y_{22} \end{bmatrix} = \begin{bmatrix} I_k & \mathbb{O} \\ \mathbb{O} & I_{n-k} \end{bmatrix},$$

or equivalently:

$$Y_{11} = U_{11}^{-1}, \quad Y_{22} = U_{22}^{-1}, \quad Y_{12} = -Y_{11}U_{12}Y_{22}.$$

Using that we can recursively compute both U and $Y = U^{-1}$.

Algorithm Chol (Recursive Cholesky factorization)

Input: *Regular, symmetric, positive definite $n \times n$ matrix A.*

1: **if** $n = 1$ **then**
2: **return** $U := [\sqrt{a_{11}}]$, $Y := \left[\sqrt{a_{11}^{-1}}\right]$
3: **end if**
4: *Decompose* $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ *such that* $A_{11} \in \mathbb{C}^{k \times k}$, *where* $k = \lfloor \frac{n}{2} \rfloor$.
5: *Compute recursively the Cholesky factorization matrix U_{11} and its inverse Y_{11} using the same algorithm for input matrix A_{11}.*
6: $U_{12} := Y_{11}^* A_{12}$
7: $T_1 := U_{12}^* U_{12}$
8: $T_2 := A_{22} - T_1$
9: *Compute recursively the Cholesky factorization matrix U_{22} and its inverse Y_{22} using the same algorithm for input matrix T_2*
10: $T_3 := -Y_{11} U_{12}$
11: $Y_{12} := T_3 Y_{22}$
12: **return** $U := \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$ *and* $Y := \begin{bmatrix} Y_{11} & Y_{12} \\ 0 & Y_{22} \end{bmatrix}$.

Recursive generalized Cholesky factorization (Singular case)

If $\mathrm{add}(n) = \mathcal{O}(n^2)$ and $\mathrm{mul}(n) = \Theta(n^{2+\epsilon})$ for $0 < \epsilon < 1$, then $\mathrm{Chol}(n) = \Theta(\mathrm{mul}(n)) = \Theta(n^{2+\epsilon})$.

If the matrix $A \in \mathbb{C}^{n\times n}$ is singular, Algorithm **Chol** will crash at step 2 for $a_{11}^{-1} = 0$.

Algorithm GenChol (Recursive generalized Cholesky factorization)

Input: *Positive semi-definite, possibly singular, matrix* $A \in \mathbb{C}^{n\times n}$.

1: ***if*** $n = 1$ ***then***
2:     ***if*** $a_{11} \neq 0$ ***then***
3:         **return** $U := [\sqrt{a_{11}}]$, $Y := \left[\sqrt{a_{11}^{-1}}\right]$
4:     ***else***
5:         **return** $U := [0]$, $Y := [0]$
6:     ***end if***
7: ***end if***
8: *Continue with steps 4-12 of Algorithm* **Chol**

Theorem (Main theorem; Petković, Stanimirović 2009)

Consider a positive semi-definite (possibly singular) matrix $A \in R^{n\times n}$. Output matrix U in Algorithm **GenChol** *satisfies $A = U^*U$. Moreover the output matrix Y is $\{1,2,3\}$ inverse of the matrix U, matrix UY is diagonal and all its entries on main diagonal are equal to 0 or 1.*

Lemma (Strassen 1969)

Let $A \in \mathbb{C}^{n\times n}$ be regular matrix partitioned as

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad A_{11} \in \mathbb{C}^{k\times k}$$

and let A_{11} is regular. Then holds

$$A^{-1} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}S^{-1} \\ -S^{-1}A_{21}A_{11}^{-1} & S^{-1} \end{bmatrix}.$$

Algorithm RapidMP (Computing the MP inverse in matrix multiplication complexity, based on the generalized Cholesky factorization)

Input: *Matrix* $A \in \mathbb{C}^{m\times n}$.

1: $A' := A^*A$
2: *Find generalized Cholesky factorization* $A' = U^*U$ *of matrix* A' *using Algorithm* **GenChol**.
3: *Obtain the matrix* L^* *by dropping zero rows from matrix* U.
4: $T := L^*L$
5: *Find the inverse* $M := T^{-1}$.
6: **return** $A^\dagger := LM^2L^*A^*$.

Time complexity: $\Theta(\mathrm{mul}(n))$, if $\mathrm{mul}(n) = \mathcal{O}(n^{2+\epsilon})$ (Strassen, Coppersmith and Winograd,...).

Similarly we can compute arbitrary $\{1,2,3\}$, $\{1,2,4\}$, $\{2,3\}_s$ i $\{2,4\}_s$ inverses (Stanimirović, Tasić 2007).

Implementation and testing results

Algorithms are implemented in programming package MATHEMATICA 6.0. Time complexity of provided implementation of Algorithm **CholGen** and Algorithm **RapidMP** is $\mathcal{O}(n^3)$.

n	CholGen	CholPivot
16	0.0047	0.0031
23	0.016	0.023
32	0.0256	0.0171
45	0.031	0.047
64	0.042	0.1232
90	0.078	0.328
128	0.1498	0.9207
180	0.265	2.481
256	3.24	10.5428
362	8.83	348.6
512	25.9895	1959.15

$\mathrm{rank} A = n/2$

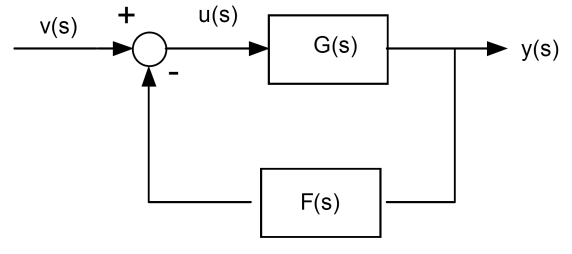
n	CholGen	CholPivot
16	0.0	0.0
23	0.003	0.0062
32	0.0094	0.0188
45	0.0094	0.0436
64	0.0378	0.1308
90	0.0346	0.3306
128	0.078	0.9422
180	0.195	2.5895
256	0.5616	7.9872
362	1.17	197.622
512	12.32	983.4

$\mathrm{rank} A = n/10$

Outline of the talk

Motivation

Design of the automatic control systems such as linear feedback system.



$$V(s) = \int_0^{+\infty} v(t)e^{-st}dt, \quad Y(s) = \int_0^{+\infty} y(t)e^{-st}dt, \quad H(s)V(s) = Y(s).$$

Problem: Determine if there exists feedback block, described by rational transfer matrix $F(s) \in \mathbb{C}^{n\times m}(s)$ such that closed loop system shown on the figure has desired transfer matrix $H(s) \in \mathbb{C}^{m\times n}(s)$?

- N.P. Karampetakis, *Computation of the generalized inverse of a polynomial matrix and applications*, Linear Algebra and its Applications, **252** (1997), 35–60.

Simple calculation yields

$$G(s)F(s)H(s) = G(s) - H(s).$$

Solution of the last equation is given by

$$F(s) = G(s)^{\dagger}(G(s) - H(s))H(s)^{\dagger} + Z(s) - G(s)^{\dagger}G(s)Z(s)H(s)H(s)^{\dagger}.$$

where $Z(s)$ is arbitrary matrix.

Theorem (Petković, Stanimirović)

The sufficient and necessary condition for solvability of the previous matrix equation is given by

$$G(s) = G(s)H(s)^{\dagger}H(s), \quad H(s) = G(s)G(s)^{\dagger}H(s).$$

Generalized inverses of rational and polynomial matrices

Previous results: Karampetakis 1997a, 1997b, 1997c; Stanimirović and Karampetakis 2000; Karampetakis and Tzekis 2001; Stanimirović 2003; Stanimirović and Tasić 2001, 2003, 2004.

Methods based on:

- Leverrier-Faddeev method,
- Greville's partitioning method.

Definition

For a given polynomial matrix $A(s) \in \mathbb{F}[s]^{n\times m}$ its maximal degree (degree) is defined as the maximal degree of its elements

$$\deg A(s) = \max\{\mathrm{dg}(A(s))_{ij} \mid 1 \le i \le n, 1 \le j \le m\}.$$

Definition

The degree matrix corresponding to $A(s) \in \mathbb{F}[s]^{n\times m}$ is the matrix defined by $\mathrm{dg}A(s) = [\mathrm{dg}A(s)_{ij}]_{m\times n}$.

Constant and rational matrices

Algorithm LFD (Leverrier-Faddeev method for computing Drazin inverse)

Input: *Constant, rational or polynomial $n \times n$ matrix A.*

$a_0 := 1$
$A_0 := \mathbb{O}$
$B_0 := I_n$
for $i := 1$ *to* n **do**
    $A_i := AB_{i-1}$
    $a_i := -\operatorname{tr}(A_i)/i$
    $B_i := A_i + a_i I_n$
end for
$k := \max\{i \mid a_i \neq 0, i = 0, \ldots, n\}$
$t := \min\{i \mid B_i = \mathbb{O}, i = 0, \ldots, n\}$
$r := t - k$
return $A^D := (-1)^{r+1} a_k^{-r-1} A^r B_{k-1}^{r+1}$

Ji 2002; Karampetakis, Stanimirović, Tasić 2007; Stanimirović, Tasić 2004;

$$A(s) = A_0 + A_1 s + \cdots + A_{q-1}s^{q-1} + A_q s^q = \sum_{i=0}^{q} A_i s^i,$$

Algorithm LFDPoly (Computing the Drazin inverse $A(s)^D$ of polynomial matrix)

Input: *The sequence of $n \times n$ constant matrices $\{A_0, A_1, \ldots, A_q\}$.*

1: $B_{0,0} := I_n, \quad B_{0,j} := \mathbb{O} \quad \forall j \in \mathbb{N}$
2: $B_{i,j} := \mathbb{O}, \quad i=0,\ldots,n-1,\ j=iq+1,\ldots,(n-1)q.$
3: $A_j := \mathbb{O}, \quad j = q+1,\ldots,nq$
4: **for** $i := 0$ *to* $n-1$ **do**
5: **for** $j := 0$ *to* $(i+1)q$ **do**
6: $a_{i+1,j} := -\frac{1}{i+1}\mathrm{Tr}\left(\sum_{l=0}^{j} A_{j-l}B_{i,l}\right), \quad B_{i+1,j} := \sum_{l=0}^{j} A_{j-l}B_{i,l} + a_{i+1,j}I_n$
7: **end for**
8: **end for**
9: $t := \max\{i \mid (\exists j \in \{0,1,\ldots,nq\})a_{i,j} \neq 0\}$
10: $r := \min\{i \mid B_{i,j} = \mathbb{O}, i = 0,\ldots,n, j = 0,1,\ldots,nq\}$
11: $k := r - t$
12: **return** $A(s)^D := (-1)^{k+1}\left(\sum_{j=0}^{tq} a_{t,j}s^j\right)^{-k-1}\left(\sum_{i=0}^{q} A_i s^i\right)^k\left(\sum_{l=0}^{(t-1)q} B_{t-1,l}s^l\right)^{k+1}.$

Main theorem

- M.D. PETKOVIĆ, P.S. STANIMIROVIĆ, *Interpolation algorithm for computing Drazin inverse of polynomial matrices*, Linear Algebra and its Applications, **422** (2007), 526–539.

Denote by k^A, B_i^A, a_i^A and t_i^A values of k, B_i, a_i and t computed by Algorithm **LFD** for an input matrix A.

Lemma

Let A be $n \times n$ constant, rational or polynomial matrix. Then holds

(a) $B^A_{t^A+i} = 0$ *for all* $i = 0, \dots, n - t^A - 1$,

(b) $B^A_{k^A+i-1} = A^{i-1}\left(AB^A + a^A I_n\right)$ *for all* $i = 1, \dots, t^A - k^A$,

(c) $a^A_{t^A+i} = 0$ *for each* $i = 0, \dots, n - t^A$ *or equivalently* $k^A \leq t^A$.

(d) *If* $A = A(s) \in \mathbb{R}[s]^{n\times n}$ *then holds* $\deg B_i(s) \leq i \cdot \deg A(s)$ *and* $\mathrm{dg} a_i(s) \leq i \cdot \deg A(s)$ *for all* $i = 0, \dots, n$.

Let $\kappa = k^{A(s)}$ and $\tau = t^{A(s)}$.

Theorem (Main theorem; Petković, Stanimirović 2007)

Let $A(s) \in \mathbb{R}[s]^{n\times n}$ and $d = \deg A(s)$. Let s_i, $i = 0, \dots, n \cdot d$ be any pairwise different real numbers. Then the following statements are valid:

- **(a)** *For $f(j) = \max\{t^{A(s_i)} \mid i = 0, \dots, j \cdot d\}$, $j = 1, \dots, n$, τ is the unique number satisfying $\tau = f(\tau)$, and also holds $\tau = f(n)$.*
- **(b)** $\kappa = \max\{k^{A(s_i)} \mid i = 0, \dots, \tau \cdot d\}$.
- **(c)** *Polynomial matrix $B^{A(s)}$ and polynomial $a^{A(s)}$ can be computed using the set of constant matrices $B^{A(s_i)}$ and values $a^{A(s_i)}$, $i = 0, \dots, \kappa \cdot d$ respectively.*

Algorithm LFDint (Interpolation alg. for computing Drazin inverse of polynomial matrix $A(s)$)

Input: *Matrix* $A(s) \in \mathbb{R}[s]^{n \times n}$

$d := \deg A(s)$; $d' := n \cdot d$; $i := -1$; $\tau := -1$
Select distinct base points $s_0, s_1, \cdots, s_{d'} \in \mathbb{R}$
repeat
    $i := i + 1$; $A_i := A(s_i)$
    Apply Algorithm **LFD** *on the input matrix* A_i*, without executing return step (step 12)*
    $\kappa_i := k^{A_i}$; $\tau_i := t^{A_i}$; $B_i' := B^{A_i}_{\kappa_i - 1}$; $a_i' := a^{A_i}_{\kappa_i}$; $\tau := \max\{\tau_i, \tau\}$
until $(i = \tau d)$ or $(i = d')$
$\kappa := \max\{\kappa_i \mid i = 0, \ldots, \tau \cdot d\}$
if $\kappa = 0$ ***then***
    return $A^D(s) := \mathbb{O}$
else
    for $i = 0$ *to* $\kappa \cdot d$ ***do***
        $$B_i := \begin{cases} A_i^{\kappa - \kappa_i - 1}\left(A_i B_i' + a_i' I_n\right), & \kappa > \kappa_i \\ B_i' & \kappa = \kappa_i \end{cases}$$
        $$a_i := \begin{cases} 0, & \kappa > \kappa_i \\ a_i', & \kappa = \kappa_i \end{cases}$$
    end for
    Interpolate polynomial $a_\kappa(s)$ *and matrix polynomial* $B_{\kappa-1}(s)$ *using pairs* (s_i, a_i) *and* (s_i, B_i), $i = 0, \ldots, \kappa \cdot d$ *as base points.*
    return $A^D(s) := (-1)^{r+1} a_\kappa(s)^{-r-1} A(s)^r B_{\kappa-1}(s)^{r+1}$
end if

Time complexity of Algorithm **LFD** is $\mathcal{O}(n^4)$ for constant and $\mathcal{O}(n^5 d^2)$ for polynomial matrices ($d = \deg A(s)$).

Time complexity of Algorithm **LFDint** is $\mathcal{O}(n^4 d'^2 + n^5 d')$.

$\deg A$	Alg **LFD**	Alg **LFDint**
5	4.475	1.7188
6	6.0092	2.3158
7	7.6874	2.9034
8	9.5314	3.6876
9	11.85	4.5624
10	14.1814	5.4844
11	16.5938	6.5156
12	19.516	7.6904
13	22.5784	8.9282
14	26.0284	10.3498
15	29.6314	11.9126

$sp_1(A) = sp_2(A) = 1$
$n = 10$, $\mathrm{rank} A = 8$

$\deg A$	Alg **LFD**	Alg **LFDint**
5	2.615	1.625
6	3.604	2.182
7	4.407	2.765
8	5.447	3.479
9	7.020	4.318
10	7.995	5.208
11	9.370	6.182
12	10.510	7.234
13	12.844	8.489
14	14.161	9.791
15	15.937	11.197

$sp_1(A) = sp_2(A) = 0.7$
$n = 10$, $\mathrm{rank} A = 8$

Interpolation methods for other generalized inverses

Algorithm LFGen (General Leverrier-Faddeev type algorithm for computing various generalized inverses)

Input: *Constant, rational or polynomial matrices* $R, T \in \mathbb{C}^{n\times m}$ *and positive integer* $e \in \mathbb{N}$.

$a_0 := 1$
$A_0 := \mathbb{O}$
$B_0 := I_n$
for $i := 1$ *to* n ***do***
    $A_i := TR^* B_{i-1}$
    $a_i := -\operatorname{tr}(A_i)/i$
    $B_i := A_i + a_i I_n$
end for
$k := \max\{i \mid a_i \neq 0, i = 0, \ldots, n\}$
if $k = 0$ ***then***
    return $X_e := \mathbb{O}$
else
    return $X_e := (-1)^e a_k^{-e} R^* B_{k-1}^e$
end if

Theorem (Stanimirović 2003)

Let A be $n \times m$ constant, rational or polynomial matrix and $A = PQ$ its full rank factorization. The following statements are valid:

(1) *If $R = T = A$ there holds $X_1 = A^\dagger$.*

(2) *If $m = n$, $R^* = A^l$, $T = A$ and $l \geq \operatorname{ind}(A)$ there holds $X_1 = A^D$.*

(3) *If $T = A$ and $n > m = \mathrm{rank} A$ for arbitrary R such that AR^* is invertible there holds $X_1 = A_R^{-1}$.*

(4) *If $m = n$, $e = l + 1$, $TR^* = A$, $R^* = A^l$ and $l \geq \operatorname{ind}(A)$ then $X_e = A^D$.*

(5) *For $m = n$, $e = 1$, $T = R^* = A^l$ and $l \geq \operatorname{ind}(A)$ we have $X_1 = (A^D)^l$.*

(6) *$X_1 \in A\{2\}$ if and only if*
$T = A, R = GH, G \in \mathbb{C}^{n\times t}, H \in \mathbb{C}^{t\times m}, \mathrm{rank} HAG = t$.

(7) *$X_1 \in A\{1,2\}$ if and only if*
$T = A, R = GH, G \in \mathbb{C}^{n\times r}, H \in \mathbb{C}^{r\times m}, \mathrm{rank} HAG = r = \mathrm{rank} A$.

(8) *$X_1 \in A\{1,2,3\}$ if and only if*
$T = A, R = GP^*, G \in \mathbb{C}^{n\times r}, \mathrm{rank} P^*AG = r = \mathrm{rank} A$.

Interpolation methods for other generalized inverses

- P.S. Stanimirović, *A finite algorithm for generalized inverses of polynomial and rational matrices*, Applied Mathematics and Computation, **144** (2003), 199–214.

Interpolation algorithm for Moore-Penrose and other various generalized inverses:

- P.S. Stanimirović, M.D. Petković, *Computation of generalized inverses of polynomial matrices by interpolation*, Applied Mathematics and Computation, **172/1** (2006), 508–523.
- M.D. Petković, P.S. Stanimirović, *Interpolation algorithm of Leverrier-Faddeev type for polynomial matrices*, Numerical Algorithms, **42** (2006), 345–361.

Computation of rank and index of polynomial matrix.

Partitioning method

$$A = [a_1\ a_2\ \cdots a_n], \quad A_k = [A_{k-1}\ a_k], \quad A = A_n$$

Algorithm PartMP (Partitioning method for computing Moore-Penrose inverse, Greville 1960)

Input: *Constant, rational or polynomial matrix* $A \in \mathbb{C}^{m\times n}$.

$A_1^\dagger := a_1^\dagger := \begin{cases} (a_1^* a_1)^{-1} a_1^*, & a_1 \neq \mathbb{O} \\ \mathbb{O}, & a_1 = \mathbb{O} \end{cases}$

for $k := 2$ *to* n **do**

  $d_k := A_{k-1}^\dagger a_k$

  $c_k := a_k - A_{k-1} d_k$

  if $c_k \neq \mathbb{O}$ ***then***

    $b_k^* := (c_k^* c_k)^{-1} c_k^*$

  else

    $b_k^* := (1 + d_k^* d_k)^{-1} d_k^* A_{k-1}^\dagger$

  end if

  $A_k^\dagger := \begin{bmatrix} A_{k-1}^\dagger - d_k b_k^* \\ b_k^* \end{bmatrix},$

end for

return $A^\dagger := A_n^\dagger$

- P.S. Stanimirović, M.B. Tasić, *Partitioning method for rational and polynomial matrices*, Applied Mathematics and Computation, **155** (2004), 137–163.
- M.D. Petković, P.S. Stanimirović, *Symbolic computation of the Moore-Penrose inverse using partitioning method*, International Journal of Computer Mathematics, 82(March 2005), pp. 355–367.

Let $s_3 = \overline{s_2}$ and $s_4 = \overline{s_1}$.

$$A(s_1, s_2) = A(S) = \sum_{j_1=0}^{q_1} \sum_{j_2=0}^{q_2} \sum_{j_3=0}^{q_3} \sum_{j_4=0}^{q_4} A_{j_1,j_2,j_3,j_4} s_1^{j_1} s_2^{j_2} s_3^{j_3} s_4^{j_4} = \sum_{J=0}^{Q} A_J S^J$$

Main theorem

Theorem (Main theorem; Petković, Stanimirović 2004)

Let $A(S) \in \mathbb{C}^{m\times n}[S]$. The MP inverse $A_i^\dagger \in \mathbb{C}^{i\times m}[S]$ of the first i columns in A is of the form

$$A_i^\dagger(S) = A_i^\dagger(S) = \frac{X_i(S)}{y_i(S)} = \frac{\sum_{J=0}^{Q_i} X_{i,J} S^J}{\sum_{J=0}^{P_i} y_{i,J} S^J}$$

where $P_i = Q_i + Q$ and $X_i \in \mathbb{C}^{i\times m}[S]$, $y_i \in \mathbb{C}[S]$ can be computed from X_{i-1}, y_{i-1}, A_{i-1} and A_i using exact recurrence relations.

Algorithm PartMPpol2 (Partitioning method for two-variable polynomial matrix)

Input: *Two-variable polynomial matrix* $A(S) \in \mathbb{C}^{m\times n}[S]$.

$X_{1,J} := a^*_{1,J}$, *for every* $0 \le J \le Q_1 = Q$

$y_{1,J} := \sum_{K=0}^{J} a^*_{i,J-\overline{K}} a_{i,K}$, *for every* $0 \le J \le P_1 = Q + \overline{Q}$

for $i := 2$ *to* n ***do***

  $d_{i,J} := \sum_{K=0}^{J} X_{i-1,J-K} A_{i,K}$, *for every* $0 \le J \le \deg d_i = Q_{i-1} + Q$

  $c_{i,J} := \sum_{K=0}^{J} (y_{i-1,J-K} a_{i,K} - A_{i-1,J-K} d_{i,K})$, *for every* $0 \le J \le \deg c_i = Q_{i-1} + 2Q$.

  if *there exists* $c_{i,J} \neq 0$ ***then***

    $w_{i,J} := \sum_{K=0}^{J} c_{i,J-K} y^*_{i-1,\overline{K}}$, *for every* $0 \le J \le \deg w_i = Q_{i-1} + \overline{Q_{i-1}} + 2Q + \overline{Q}$

    $v_{i,J} := \sum_{K=0}^{J} c^*_{i,J-K} c_{i-1,\overline{K}}$, *for every* $0 \le J \le \deg v_i = \deg w_i + \overline{Q}$

  else

    $w_{i,J} := \sum_{K=0}^{\overline{J}} X^*_{i-1,J-K} d_{i,\overline{K}}$, *for every* $0 \le J \le \deg w_i = Q_{i-1} + \overline{Q_{i-1}} + Q + \overline{Q}$

    $v_{i,J} = \sum_{K=0}^{\overline{J}} (y^*_{i-1,J-K} y_{i,\overline{K}} + d^*_{i,J-K} d_{i,\overline{K}})$, *for every* $0 \le J \le \deg v_i = Q_{i-1} + \overline{Q_{i-1}} + Q + \overline{Q}$

  end if

  $\Theta_{i,J} := \sum_{K=0}^{J} v^*_{i,K} X_{i-1,J-\overline{K}} - d_{i,J-\overline{K}} w^*_{i,K}$, *for every* $0 \le J \le \deg \Theta_i = \overline{\deg w_i} + Q + Q_{i-1}$

  $\phi_{i,J} := \sum_{K=0}^{J} v^*_{i,K} y_{i-1,J-\overline{K}}$, *for every* $0 \le J \le \deg \phi_i = \overline{\deg v_i} + Q + Q_{i-1}$

  $X_{i,J} := \begin{bmatrix} \sum_{K=0}^{J} v^*_{i,K} \Theta_{i,J-\overline{K}} \\ \sum_{K=0}^{J} \phi_{i,J-\overline{K}} w^*_{i,K} \end{bmatrix}$, *for every* $0 \le J \le Q_i = Q_{i-1} + Q + \overline{\deg w_i} + \overline{\deg v_i}$

  $y_{i,J} := \sum_{K=0}^{J} v^*_{i,K} \phi_{i,J-\overline{K}}$, *for every* $0 \le J \le P_i = Q_i + \overline{Q}$

end for

Partitioning method for weighted Moore-Penrose inverse

- G.R. Wang, Y.L.Chen, *A recursive algorithm for computing the weighted Moore-Penrose inverse $A^{\dagger}_{MN}$*, Journal of Computational mathematics, **4** (1986), 74–85.

Polynomial and rational matrices:

- M.B. Tasić, P.S. Stanimirović, M.D. Petković, *Symbolic computation of weighted Moore-Penrose inverse using partitioning method*, Applied Mathematics and Computation, **189** (2007) 615–640.
- M.D. Petković, P.S. Stanimirović, M.B. Tasić, *Effective partitioning method for computing weighted Moore-Penrose inverse*, Computers & Mathematics with Applications, **55** (2008), Issue 8, 1720–1734.

Some more results

- P.S. Stanimirović, M.B. Tasić, K. Vu, *Extensions of Faddeev's algorithms to polynomial matrices*, Applied Mathematics and Computation, **214** (2009), 246–258.
- P.S. Stanimirović, M.B. Tasić, *Drazin inverse of one-variable polynomial matrices*, Filomat, Niš **15** (2001), 71–78.
- M.B. Tasić, P.S. Stanimirović, *Symbolic and recursive computation of different types of generalized inverses*, Applied Mathematics and Computation, **199** (2008), 349–367.
- M.D. Petković, P.S. Stanimirović, *Partitioning method for two-variable rational and polynomial matrices*, Mathematica Balcanica **19** (2005), 185–194.

Thanks for attention!