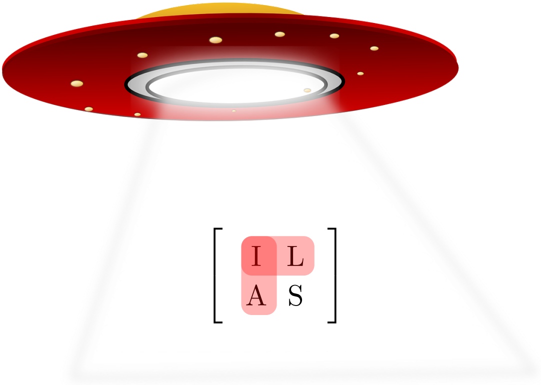
Principal pivot transforms, structured matrices, and matrix equations

Federico Poloni

University of Pisa

ILAS 2019: Linear Algebra Without Borders
Rio de Janeiro, July 2019

What if Martians had linear algebra?



They would have the same underlying results, but possibly in an ‘alien’ notation or format: they may not have the same primitives such as linear maps, factorization, or even equal signs.

Principal pivot transforms feel a lot like a tool from a different world.

Principal pivot transforms

Definition

Let $A \in \mathbb{R}^{n\times n}$, $\mathfrak{s} = \{1 : k\}$ (Fortran/Matlab notation), and define (when A_{11} is invertible)

$$\operatorname{ppt}_{\mathfrak{s}}(A) = \operatorname{ppt}_{\mathfrak{s}}(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}) := \begin{bmatrix} -A_{11}^{-1} & A_{11}^{-1}A_{12} \\ A_{21}A_{11}^{-1} & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix}.$$

Several classical linear algebra objects: inverses, linear system solutions, Schur complements; packaged in an unusual form.

Principal pivot transforms

Definition

Let $A \in \mathbb{R}^{n\times n}$, $\mathfrak{s} = \{1 : k\}$ (Fortran/Matlab notation), and define (when A_{11} is invertible)

$$\operatorname{ppt}_{\mathfrak{s}}(A) = \operatorname{ppt}_{\mathfrak{s}}\left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\right) := \begin{bmatrix} -A_{11}^{-1} & \pm A_{11}^{-1}A_{12} \\ \pm A_{21}A_{11}^{-1} & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix}.$$

Several classical linear algebra objects: inverses, linear system solutions, Schur complements; packaged in an unusual form.

Technical detail: we will allow for minus signs on the rows of the $(2,1)$ block, and columns of the $(1,2)$ block.

Signs are important to get symmetry right, but we will not be concerned with them in this talk.

PPTs with general indices

If $\mathfrak{s} \subset \{1, 2, \ldots, n\}$ is not $1 : k$, we take the same definition but with the first block to mean "the entries in $\mathfrak{s}$": to get

$$\operatorname{ppt}_{\{1,3,4\}}\left(\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}\right),$$

replace the dark block with minus its inverse, the white block with the Schur complement, and multiply by the inverse the rows/columns in the light block.

Some would write it

$$\begin{bmatrix} B[\mathfrak{s},\mathfrak{s}] & B[\mathfrak{s},\mathfrak{s}'] \\ B[\mathfrak{s}',\mathfrak{s}] & B[\mathfrak{s}',\mathfrak{s}'] \end{bmatrix} = \begin{bmatrix} -A[\mathfrak{s},\mathfrak{s}]^{-1} & A[\mathfrak{s},\mathfrak{s}]^{-1}A[\mathfrak{s},\mathfrak{s}'] \\ A[\mathfrak{s}',\mathfrak{s}]A[\mathfrak{s},\mathfrak{s}]^{-1} & A[\mathfrak{s}',\mathfrak{s}'] - A[\mathfrak{s}',\mathfrak{s}]A[\mathfrak{s},\mathfrak{s}]^{-1}A[\mathfrak{s},\mathfrak{s}'] \end{bmatrix}.$$

Swapping variables

Review paper [Tsatsomeros, 2000]: PPTs appear in various fields. One way to think about them: $Ax = b$ holds iff

$$\begin{bmatrix} -A_{11}^{-1} & A_{11}^{-1}A_{12} \\ A_{21}A_{11}^{-1} & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix} \begin{bmatrix} b_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -x_1 \\ b_2 \end{bmatrix}.$$

PPTs "swap" some of the unknowns with right-hand sides.

Elementary PPTs

When the block to be inverted is 1×1, a PPT takes $O(n^2)$ operations: most of it is a rank-1 update of a $(n-1) \times (n-1)$ submatrix.

$$\left[\begin{array}{c|c} -a_{11}^{-1} & a_{11}^{-1} a_{12} \\ \hline a_{21} a_{11}^{-1} & A_{22} - a_{21} a_{11}^{-1} a_{12} \end{array}\right],$$

$$\operatorname{ppt}_{\{4\}}\left(\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}\right).$$

Quiz: a mysterious alien algorithm

Standard algorithm on every linear algebra textbook published on Mars:

Tnhff-Wbeqna algorithm

Start from $A \in \mathbb{R}^{n \times n}$, and perform elementary PPTs on the entries $1, 2, 3, \ldots, n$ in sequence. (Actually, in any order at your choice.)

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}$$

- What does this algorithm compute?
- What do we call it on Earth?

Quiz: a mysterious alien algorithm

Standard algorithm on every linear algebra textbook published on Mars:

Tnhff-Wbeqna algorithm

Start from $A \in \mathbb{R}^{n \times n}$, and perform elementary PPTs on the entries $1, 2, 3, \ldots, n$ in sequence. (Actually, in any order at your choice.)

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}$$

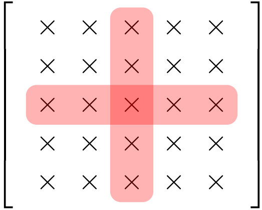
- What does this algorithm compute?
- What do we call it on Earth?

Quiz: a mysterious alien algorithm

Standard algorithm on every linear algebra textbook published on Mars:

Tnhff-Wbeqna algorithm

Start from $A \in \mathbb{R}^{n\times n}$, and perform elementary PPTs on the entries $1, 2, 3, \ldots, n$ in sequence. (Actually, in any order at your choice.)



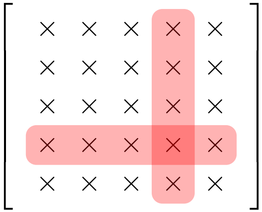
- What does this algorithm compute?
- What do we call it on Earth?

Quiz: a mysterious alien algorithm

Standard algorithm on every linear algebra textbook published on Mars:

Tnhff-Wbeqna algorithm

Start from $A \in \mathbb{R}^{n \times n}$, and perform elementary PPTs on the entries $1, 2, 3, \ldots, n$ in sequence. (Actually, in any order at your choice.)



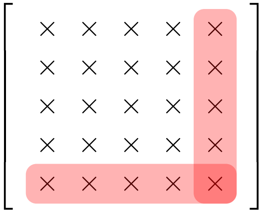
- What does this algorithm compute?
- What do we call it on Earth?

Quiz: a mysterious alien algorithm

Standard algorithm on every linear algebra textbook published on Mars:

Tnhff-Wbeqna algorithm

Start from $A \in \mathbb{R}^{n \times n}$, and perform elementary PPTs on the entries $1, 2, 3, \ldots, n$ in sequence. (Actually, in any order at your choice.)



- What does this algorithm compute?
- What do we call it on Earth?

?

?

Gauss–Jordan algorithm

- Start from $\begin{bmatrix} A & -I \end{bmatrix}$.
- Perform row elementary operations to transform it into $\begin{bmatrix} I & X \end{bmatrix}$.
- Then, $X = -A^{-1}$.

$$\left[\begin{array}{ccccc|ccccc} \times & \times & \times & \times & \times & -1 & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & 0 & -1 & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & 0 & 0 & -1 & 0 & 0 \\ \times & \times & \times & \times & \times & 0 & 0 & 0 & -1 & 0 \\ \times & \times & \times & \times & \times & 0 & 0 & 0 & 0 & -1 \end{array}\right]$$

- Each step is an elementary PPT;
- We store only the "active" part of the matrix at each step, keeping columns $\mod n$.
- Cost: $2n^3$, exactly like `inv`.

Gauss–Jordan algorithm

- Start from $\begin{bmatrix} A & -I \end{bmatrix}$.
- Perform row elementary operations to transform it into $\begin{bmatrix} I & X \end{bmatrix}$.
- Then, $X = -A^{-1}$.

$$\left[\begin{array}{ccccc|ccccc} 1 & \times & \times & \times & \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times & \times & -1 & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times & \times & 0 & -1 & 0 & 0 \\ 0 & \times & \times & \times & \times & \times & 0 & 0 & -1 & 0 \\ 0 & \times & \times & \times & \times & \times & 0 & 0 & 0 & -1 \end{array}\right]$$

- Each step is an elementary PPT;
- We store only the "active" part of the matrix at each step, keeping columns $\mod n$.
- Cost: $2n^3$, exactly like `inv`.

Gauss–Jordan algorithm

- Start from $\begin{bmatrix} A & -I \end{bmatrix}$.
- Perform row elementary operations to transform it into $\begin{bmatrix} I & X \end{bmatrix}$.
- Then, $X = -A^{-1}$.

$$\left[\begin{array}{ccccc|ccccc} 1 & 0 & \times & \times & \times & \times & \times & 0 & 0 & 0 \\ 0 & 1 & \times & \times & \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & \times & \times & \times & \times & \times & -1 & 0 & 0 \\ 0 & 0 & \times & \times & \times & \times & \times & 0 & -1 & 0 \\ 0 & 0 & \times & \times & \times & \times & \times & 0 & 0 & -1 \end{array}\right]$$

- Each step is an elementary PPT;
- We store only the "active" part of the matrix at each step, keeping columns $\mod n$.
- Cost: $2n^3$, exactly like `inv`.

Gauss–Jordan algorithm

- Start from $\begin{bmatrix} A & -I \end{bmatrix}$.
- Perform row elementary operations to transform it into $\begin{bmatrix} I & X \end{bmatrix}$.
- Then, $X = -A^{-1}$.

$$\left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & \times & \times & \times & \times & \times & 0 & 0 \\ 0 & 1 & 0 & \times & \times & \times & \times & \times & 0 & 0 \\ 0 & 0 & 1 & \times & \times & \times & \times & \times & 0 & 0 \\ 0 & 0 & 0 & \times & \times & \times & \times & \times & -1 & 0 \\ 0 & 0 & 0 & \times & \times & \times & \times & \times & 0 & -1 \end{array}\right]$$

- Each step is an elementary PPT;
- We store only the “active” part of the matrix at each step, keeping columns $\mod n$.
- Cost: $2n^3$, exactly like `inv`.

Gauss–Jordan algorithm

- Start from $\begin{bmatrix} A & -I \end{bmatrix}$.
- Perform row elementary operations to transform it into $\begin{bmatrix} I & X \end{bmatrix}$.
- Then, $X = -A^{-1}$.

$$\left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & \times & \times & \times & \times & \times & 0 \\ 0 & 1 & 0 & 0 & \times & \times & \times & \times & \times & 0 \\ 0 & 0 & 1 & 0 & \times & \times & \times & \times & \times & 0 \\ 0 & 0 & 0 & 1 & \times & \times & \times & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times & \times & \times & \times & -1 \end{array}\right]$$

- Each step is an elementary PPT;
- We store only the "active" part of the matrix at each step, keeping columns $\mod n$.
- Cost: $2n^3$, exactly like `inv`.

Gauss–Jordan algorithm

- Start from $\begin{bmatrix} A & -I \end{bmatrix}$.
- Perform row elementary operations to transform it into $\begin{bmatrix} I & X \end{bmatrix}$.
- Then, $X = -A^{-1}$.

$$\left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & \times & \times & \times & \times & \times \\ 0 & 1 & 0 & 0 & 0 & \times & \times & \times & \times & \times \\ 0 & 0 & 1 & 0 & 0 & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 1 & 0 & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 1 & \times & \times & \times & \times & \times \end{array}\right]$$

- Each step is an elementary PPT;
- We store only the "active" part of the matrix at each step, keeping columns $\mod n$.
- Cost: $2n^3$, exactly like `inv`.

What is going on

Given $A \in \mathbb{R}^{n\times n}$ and $\mathfrak{s} \subset \{1,2,\dots,n\}$, let $G_{\mathfrak{s}}(A)$ be the $2n \times n$ matrix with columns of $\pm I$ in positions $\mathfrak{s}$ and $n+\mathfrak{s}'$, and of A elsewhere:

$$G_{\{1,4\}}(A) := \left[\begin{array}{ccccc|ccccc} 1 & A_{12} & A_{13} & 0 & A_{15} & A_{11} & 0 & 0 & A_{14} & 0 \\ 0 & A_{22} & A_{23} & 0 & A_{25} & A_{21} & -1 & 0 & A_{24} & 0 \\ 0 & A_{32} & A_{33} & 0 & A_{35} & A_{31} & 0 & -1 & A_{34} & 0 \\ 0 & A_{42} & A_{43} & 1 & A_{45} & A_{41} & 0 & 0 & A_{44} & 0 \\ 0 & A_{52} & A_{53} & 0 & A_{55} & A_{51} & 0 & 0 & A_{54} & -1 \end{array}\right]$$

$$G_{\{2,3,4\}}(B) := \left[\begin{array}{ccccc|ccccc} B_{11} & 0 & 0 & 0 & B_{15} & -1 & B_{12} & B_{13} & B_{14} & 0 \\ B_{21} & 1 & 0 & 0 & B_{25} & 0 & B_{22} & B_{23} & B_{24} & 0 \\ B_{31} & 0 & 1 & 0 & B_{35} & 0 & B_{32} & B_{33} & B_{34} & 0 \\ B_{41} & 0 & 0 & 1 & B_{45} & 0 & B_{42} & B_{43} & B_{44} & 0 \\ B_{51} & 0 & 0 & 0 & B_{55} & 0 & B_{52} & B_{53} & B_{54} & -1 \end{array}\right]$$

What is going on

Theorem

$G_{\mathfrak{s}_1}(A)$ and $G_{\mathfrak{s}_2}(B)$ have the same row space $\iff$ $B = \operatorname{ppt}_{\mathfrak{s}_1 \Delta \mathfrak{s}_2}(A)$.

- PPTs convert between G-matrices that have the same row space i.e., they are equivalent by row operations / left multiplication.
- For each k, one among columns k and $n+k$ is $\pm e_k$. Each PPT with $k \in \mathfrak{s}$ switches between the two positions.

Consequences:

- All sequences of PPTs that produce the same final $\mathfrak{s}$ return the same matrix.
- The only thing that matters is whether each index k is 'inverted' an even or odd number of times;
- PPTs commute one with each other.

Example Any sequence of PPTs that acts once on each k transforms $\begin{bmatrix} A & -I \end{bmatrix}$ into the equivalent matrix $\begin{bmatrix} I & -A^{-1} \end{bmatrix}$.

Symmetry

If A is symmetric, then $\operatorname{ppt}_{\mathfrak{s}}(A)$ is symmetric, too.

Clear from the definition:

$$\operatorname{ppt}_{\mathfrak{s}}(A) = \begin{bmatrix} -A_{11}^{-1} & \pm A_{11}^{-1}A_{12} \\ \pm A_{21}A_{11}^{-1} & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix}$$

Actually, here we presented the theory with symmetry in mind: a non-symmetric variant with two subsets (rows/columns) instead of one is possible.

Just for fun

A Martian proof that $(AB)^{-1} = B^{-1}A^{-1}$ using (non-symmetric) PPTs:

$$\begin{bmatrix} I & B \\ A & 0 \end{bmatrix}$$

Just for fun

A Martian proof that $(AB)^{-1} = B^{-1}A^{-1}$ using (non-symmetric) PPTs:

$$\begin{bmatrix} -I & B \\ A & -AB \end{bmatrix}$$

Just for fun

A Martian proof that $(AB)^{-1} = B^{-1}A^{-1}$ using (non-symmetric) PPTs:

$$\begin{bmatrix} -I + B(AB)^{-1}A & -B(AB)^{-1} \\ -(AB)^{-1}A & (AB)^{-1} \end{bmatrix}$$

The same PPTs in a different order:

Just for fun

A Martian proof that $(AB)^{-1} = B^{-1}A^{-1}$ using (non-symmetric) PPTs:

$$\begin{bmatrix} -I + B(AB)^{-1}A & -B(AB)^{-1} \\ -(AB)^{-1}A & (AB)^{-1} \end{bmatrix}$$

The same PPTs in a different order:

$$\begin{bmatrix} I & B \\ A & 0 \end{bmatrix}$$

Just for fun

A Martian proof that $(AB)^{-1} = B^{-1}A^{-1}$ using (non-symmetric) PPTs:

$$\begin{bmatrix} -I + B(AB)^{-1}A & -B(AB)^{-1} \\ -(AB)^{-1}A & (AB)^{-1} \end{bmatrix}$$

The same PPTs in a different order:

$$\begin{bmatrix} 0 & -A^{-1} \\ B & A^{-1} \end{bmatrix}$$

Just for fun

A Martian proof that $(AB)^{-1} = B^{-1}A^{-1}$ using (non-symmetric) PPTs:

$$\begin{bmatrix} -I + B(AB)^{-1}A & -B(AB)^{-1} \\ -(AB)^{-1}A & (AB)^{-1} \end{bmatrix}$$

The same PPTs in a different order:

$$\begin{bmatrix} 0 & -A^{-1} \\ -B^{-1} & B^{-1}A^{-1} \end{bmatrix}$$

Just for fun

A Martian proof that $(AB)^{-1} = B^{-1}A^{-1}$ using (non-symmetric) PPTs:

$$\begin{bmatrix} -I + B(AB)^{-1}A & -B(AB)^{-1} \\ -(AB)^{-1}A & (AB)^{-1} \end{bmatrix}$$

The same PPTs in a different order:

$$\begin{bmatrix} 0 & -A^{-1} \\ -B^{-1} & B^{-1}A^{-1} \end{bmatrix}$$

- We could have used symmetric PPTs and a $\left[\begin{smallmatrix} 0 & M \\ M^T & 0 \end{smallmatrix}\right]$ trick.
- Comparing products of pivots, one also gets the relation $\det(AB) = \det(A)\det(B)$.

Indefinite linear algebra

Matrices $G_{\mathfrak{s}}(A)$ are related to various matrix structures of indefinite linear algebra with the (antisymmetric) scalar product

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}.$$

For each $\mathfrak{s}$ and $M = M^T$, the rows of $G_{\mathfrak{s}}(M)$ span a Lagrangian subspace W (W equals its J-orthogonal $W^{\perp}$).
Actually, each Lagrangian W has a basis of the form $G_{\mathfrak{s}}(M)$. [Dopico Johnson '06, Mehrmann FP '12]

Structured pencils [Mehrmann FP '12]

Various structured pencils can be written analogously by stacking columns of $\pm I$ and columns of a symmetric $M = \begin{bmatrix} G & A \\ A^T & -Q \end{bmatrix}$: e.g.,

- Hamiltonian (J-skew-selfadjoint): $\begin{bmatrix} A & G \\ -Q & A^T \end{bmatrix} - \lambda \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$;
- Symplectic (J-orthogonal): $\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} - \lambda \begin{bmatrix} I & G \\ 0 & A^T \end{bmatrix}$.

Same structure, up to block swaps $\implies$ same tools can be used.

Applying row transformations to turn $\begin{bmatrix} \mathcal{A} & \mathcal{E} \end{bmatrix}$ into $\begin{bmatrix} K\mathcal{A} & K\mathcal{E} \end{bmatrix}$

$$\iff$$

Transforming $\mathcal{A} - \lambda\mathcal{E}$ into a pencil $K(\mathcal{A} - \lambda\mathcal{E})$ with same eigenvalues and right eigenvectors.

Permuted graph bases [Mehrmann FP '12]

Particularly interesting because one can obtain well-conditioned $G_{\mathfrak{s}}(M)$:

Theorem

Every Lagrangian W admits a basis $G_{\mathfrak{s}}(M)$ (with a well-chosen $\mathfrak{s}$) with

$$\max_{ij} |M_{ij}| \leq \sqrt{2}.$$

Proof: given any basis $W \in \mathbb{R}^{n \times 2n}$, among all 2^n possible locations $W_{:,\alpha}$ where we can put I, choose the one with maximal $|\det W_{:,\alpha}|$.

Bounded $M \implies$ small condition number $\kappa(G_{\mathfrak{s}}(M))$.
Well-conditioned, exactly structure-preserving basis.

Similar "bases" can be used to work with symplectic and Hamiltonian pencils.

Example

$$Z = \left[\begin{array}{cccc|cccc} 1 & \frac{11}{3} & -\frac{10}{3} & 1 & \frac{1}{6} & -2 & 0 & \frac{8}{3} \\ 0 & -\frac{7}{3} & \frac{7}{3} & -1 & \frac{2}{3} & 1 & 0 & -\frac{7}{3} \\ 1 & \frac{1}{3} & -1 & 0 & \frac{5}{6} & -1 & -1 & -1 \\ 0 & \frac{7}{3} & -\frac{7}{3} & 1 & -\frac{2}{3} & -1 & 0 & \frac{7}{3} \end{array}\right].$$

$\operatorname{Im} Z^T$ is Lagrangian. It has a basis of the form $G_{\mathfrak{s}}(M)$ with $M = M^T$ and $\max_{ij}|M_{ij}| \leq \sqrt{2}$:

$$G_{\{1,4\}}(M) = \left[\begin{array}{cccc|cccc} 1 & \frac{1}{3} & 0 & 0 & \frac{1}{2} & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & \frac{1}{3} & -1 & 0 & \frac{4}{3} \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & -\frac{4}{3} \\ 0 & \frac{4}{3} & -\frac{4}{3} & 1 & -1 & 0 & 0 & 1 \end{array}\right].$$

Remark The non-symmetric analogue (every subspace has a non-symmetric-PPT basis $G_{\mathfrak{s}}(M)$ with $\max_{ij}|M|_{ij} \leq 1$) is in [Knuth '85].

Quasi-definiteness

Another structure: quasi-definiteness. [George, Ikramov '00]

Definition

$A = A^T$ is $\mathfrak{s}$-quasi-definite ($\mathfrak{s}$-qd) if $A_{\mathfrak{s},\mathfrak{s}} \succ 0$ and $A_{\mathfrak{s}',\mathfrak{s}'} \prec 0$ (complementary blocks of opposite definiteness).

Cfr. saddle-point matrices in optimization. [Benzi, Golub, Liesen '05]

If $M = M^T \succ 0$, then $\operatorname{ppt}_{\mathfrak{s}}(M)$ exists for all $\mathfrak{s}$, and is $\mathfrak{s}$-quasi-definite.

Clear from the definition:

$$\operatorname{ppt}_{\mathfrak{s}}(M) = \begin{bmatrix} -M_{11}^{-1} & \pm M_{11}^{-1} M_{12} \\ \pm M_{21} M_{11}^{-1} & M_{22} - M_{21} M_{11}^{-1} M_{12} \end{bmatrix}$$

PPTs transform qd matrices into other qd matrices (while changing the partition).

PPTs and quasidefiniteness

Consequence (by continuity):

Suppose $M = M^T$ is $\mathfrak{s}_1$-weakly-qd ($\prec, \succ$ replaced by $\preceq, \succeq$).
Then, for each subset $\mathfrak{s}_2$, the matrix $\operatorname{ppt}_{\mathfrak{s}_2}(M)$ is $\mathfrak{s}_1 \,\Delta\, \mathfrak{s}_2$-weakly-qd (when it exists). [FP, Strabić '16]

Example:

$$\operatorname{ppt}_{\{3,4\}}\left(\begin{bmatrix} + & + & + & \times & \times \\ + & + & + & \times & \times \\ + & + & + & \times & \times \\ \times & \times & \times & - & - \\ \times & \times & \times & - & - \end{bmatrix}\right) = \begin{bmatrix} + & + & \times & + & \times \\ + & + & \times & + & \times \\ \times & \times & - & \times & - \\ + & + & \times & + & \times \\ \times & \times & - & \times & - \end{bmatrix}.$$

The index 3 "switches" from the positive semidef. part to the negative semidef. part; the index 4 does the opposite.

Factored PPTs

Weakly-qd matrices appear frequently in applications, e.g., control theory.

Symplectic $\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} - \lambda \begin{bmatrix} I & G \\ 0 & A^T \end{bmatrix}$ and Hamiltonian $\begin{bmatrix} A & G \\ -Q & A^T \end{bmatrix} - \lambda \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$ are built with columns of the quasi-definite

$$\begin{bmatrix} G & A \\ A^T & -Q \end{bmatrix} = \begin{bmatrix} BB^T & A \\ A^T & -C^T C \end{bmatrix}.$$

Often, $\mathrm{rk}(G)$ and $\mathrm{rk}(Q)$ are very small.

Can we perform PPTs while keeping the semidefinite blocks factored?

Factored PPTs

We parametrize a $\mathfrak{s}$-weakly-qd matrix with (A, B, C) such that

$$M = \begin{bmatrix} BB^T & A \\ A^T & -C^T C \end{bmatrix} =: p\left(\left[\begin{array}{c|c} B & A \\ \hline \star & C \end{array}\right]\right)$$

(assume $\mathfrak{s} = \{1, 2, \dots, k\}$ to keep blocks ordered.)

Remark A not necessarily square.

Perform an elementary PPT on entry $k \in \mathfrak{s}$, and look at the semidefinite blocks:

$$(BB^T)_{1:k-1,1:k-1} - (BB^T)_{1:k-1,k}(BB^T)^{-1}_{k,k}(BB^T)_{k,1:k-1},$$
$$-(C^T C) - (A^T)_{:,k}(BB^T)^{-1}_{k,k}(A)_{k,:}.$$

We add a rk-1 term to $C^T C \implies$ one row inserted in C.
We subtract a rk-1 term from $BB^T \implies$ one column removed from B (hope).

Factored PPTs: the formula [FP, Strabić '16]

Nice-looking formulas if we apply a Householder reflector H to insert zeros in the last row of B:

$$\mathrm{ppt}_{\{k\}}(p(\left[\begin{array}{c|c} B & A \\ \hline \star & C \end{array}\right])) = \mathrm{ppt}_{\{k\}}(p(\left[\begin{array}{c|c} B{\color{red}H} & A \\ \hline \star & C \end{array}\right]))$$

$$= \mathrm{ppt}_{\{k\}}(p(\left[\begin{array}{cc|c} B_{11} & b & A_1 \\ 0 & \beta & a \\ \hline \star & \star & C \end{array}\right])) = p(\left[\begin{array}{c|cc} B_{11} & \pm b\beta^{-1} & A_1 - b\beta^{-1}a \\ \hline \star & \beta^{-1} & \pm\beta^{-1}a \\ \star & 0 & C \end{array}\right]).$$

Surprisingly, these formulas to update the factors are very similar to a non-factored PPT.

Factored PPTs: the formula

Analogous formula for an elementary PPT with an index in the $C^T C$ block:

$$\operatorname{ppt}_{\{k+1\}}\left(p\left(\left[\begin{array}{c|c} B & A \\ \hline \star & C \end{array}\right]\right)\right) = \operatorname{ppt}_{\{k+1\}}\left(p\left(\left[\begin{array}{c|c} B & A \\ \hline \star & HC \end{array}\right]\right)\right)$$

$$= \operatorname{ppt}_{\{k+1\}}\left(p\left(\left[\begin{array}{c|cc} B & a & A_2 \\ \hline \star & \gamma & c \\ \star & 0 & C_{22} \end{array}\right]\right)\right) = p\left(\left[\begin{array}{cc|c} B & \pm a\gamma^{-1} & A_2 - a\gamma^{-1}c \\ 0 & \gamma^{-1} & \pm\gamma^{-1}c \\ \hline \star & \star & C_{22} \end{array}\right]\right).$$

Remark We switch rows/columns around between blocks, but $\left[\begin{smallmatrix} B & A \\ \star & C \end{smallmatrix}\right]$ never changes size.

Inverting quasi-semidefinite matrices

- We know how to perform factored PPTs;
- Elementary PPTs on indices $1, 2, \ldots, n$ (in any order) can be used to invert a matrix.

These two ingredients produce an algorithm to compute inverses of quasi-semidefinite matrices

$$\begin{bmatrix} BB^T & A \\ A^T & -C^T C \end{bmatrix}^{-1} = \begin{bmatrix} \hat{B}\hat{B}^T & \hat{A} \\ \hat{A}^T & -\hat{C}^T \hat{C} \end{bmatrix}.$$

Just perform n PPTs one after the other, in factored form!

Exact ranks are preserved: $(\hat{A}, \hat{B}, \hat{C})$ have the same sizes as (A^T, C^T, B^T).

Pivoting

Pivoting (i.e., reordering elementary PPTs) works the same as the classical LDL^T theory [Bunch–Parlett '71, Bunch–Kaufman '77]: at each step,

- either locate a large diagonal pivot $|M_{ii}|\dots$
- $\dots$or a 2×2 pivot with large offdiagonal $|M_{ij}|$ and smaller diagonal $|M_{ii}|, |M_{jj}|$.

But using quasi-definiteness, we can cut some corners:

- Off-diagonal entries in the blocks $BB^T, -C^T C$ are always smaller than diagonal ones;
- 2×2 pivots $P = \begin{bmatrix} \beta & \alpha \\ \overline{\alpha} & \gamma \end{bmatrix}$ have $\beta \geq 0, \gamma \leq 0$, hence there is no cancellation in $\det P = \beta\gamma - |\alpha|^2$.

Technical detail: we also need a 2×2 version of the factored update formulas.

Stability

Gauss–Jordan can be unstable for general matrices, but not for quasidefinite ones:

Theorem (backward stability) [Benner FP]

When Gauss–Jordan / successive PPTs are used to compute $X = M^{-1}$ for a quasidefinite M (not in factored form), the jth column of $\hat{X}$ is the jth column of $(M + \Delta)^{-1}$, with

$$|\Delta| \leq p(n)\mathbf{u}(|L||D||L^*| + |M||L^{-*}||L^*|).$$

Backward stable if:

1. Not too much element growth in $M = LDL^*$;
2. Not too much element growth when forming L^{-1}.

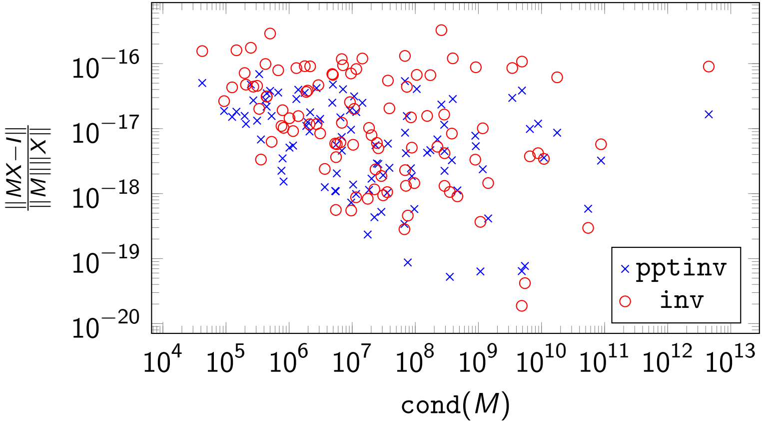
1 and 2 are related, since $D = L^{-1}ML^{-*}$.

([Peters-Wilkinson '75, Higham '97, Malyshev '00] treat LDL and GJ separately.)

Stability

What about the method with factored-form updates?

Proving stability seems challenging, but computationally residuals are as small as with `inv`. On 100 matrices with random badly-scaled A, B, C:



Application: sign function and Riccati equations

Definition

Given $A = V \operatorname{diag}(\lambda_i) V^{-1}$, define $\operatorname{sign}(A) = V \operatorname{diag}(\operatorname{sign}(\lambda_i)) V^{-1}$, where

$$\operatorname{sign}(\lambda_i) = \begin{cases} -1 & \operatorname{Re}(\lambda_i) < 0, \\ 1 & \operatorname{Re}(\lambda_i) > 0. \end{cases}$$

Theorem [Roberts, '71] Let $S = \operatorname{sign}\left(\begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix}\right)$. Then,

$\ker S + I = \operatorname{span} \begin{bmatrix} I \\ -X \end{bmatrix}$ and $\ker S - I = \operatorname{span} \begin{bmatrix} Y \\ I \end{bmatrix}$, where $X \succeq 0$ and $Y \succeq 0$ solve the Riccati equations

$$A^T X + XA + C^T C = XBB^T X,$$
$$YA^T + AY + BB^T = YC^T CY.$$

The matrix sign iteration

Matrix sign iteration

$$H_0 = H, \quad H_{k+1} = \frac{1}{2}(H_k + H_k^{-1}).$$

The iteration converges to $\lim_{k\to\infty} H_k = \operatorname{sign}(H)$.

It can be recast using weakly-qd matrices $M_k = H_k J$. [Gardiner-Laub '86].

$$M_0 = HJ, \quad M_{k+1} = \frac{1}{2}(M_k + JM_k^{-1}J).$$

PPT $\implies$ matrix sign [Benner, FP]

Algorithm

1. Start from $H_0 J = M_0 = p\left(\left[\begin{array}{c|c} B & A \\ \hline \star & C \end{array}\right]\right)$ from system data
2. Compute $M_0^{-1} = p\left(\left[\begin{array}{c|c} \hat{B} & \hat{A} \\ \hline \star & \hat{C} \end{array}\right]\right)$
3. Form $M_1 = \frac{1}{2}(M_0 + J M_0^{-1} J) = p\left(\left[\begin{array}{c|c} \frac{1}{\sqrt{2}}\begin{bmatrix} B & \hat{C}^T \end{bmatrix} & \frac{1}{2}(A + \hat{A}^T) \\ \hline \star & \frac{1}{\sqrt{2}}\begin{bmatrix} C \\ \hat{B}^T \end{bmatrix} \end{array}\right]\right)$
4. Optionally, “compress” (`rrqr`) $\begin{bmatrix} B & \hat{C}^T \end{bmatrix}$ and $\begin{bmatrix} C \\ \hat{B}^T \end{bmatrix}$
5. Repeat: $M_2, M_3, M_4, \dots$ until convergence to $\operatorname{sign}(H_0)J$.

Uses of the matrix sign

This algorithm computes

$$\operatorname{sign}(H_0) = \begin{bmatrix} A_s & B_s B_s^T \\ C_s^T C_s & -A_s^T \end{bmatrix}$$

directly in factored form (without forming Gram matrices and refactoring them as in [Benner–Ezzatti–Quintana Ortí–Remón '14]).

What do we do with it?

- B_s, C_s used directly in applications in model reduction [Wortelboer, '94]
- Solutions to CAREs from $\ker(\operatorname{sign}(H_0) \pm I)$

It is well-established that often $X = ZZ^T, Y = WW^T$ have low numerical rank (see e.g. [Benner, Bujanović '16]).

Can we compute them directly in factored form?

Think like an alien

Idea Try to see everything as PPTs / Schur complements.

Cayley transform via PPTs [Benner, FP]

Given $H = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix}$, we can compute its Cayley transform

$$(H - I)^{-1}(H + I) = \begin{bmatrix} I & B_c B_c^T \\ 0 & A_c^T \end{bmatrix}^{-1} \begin{bmatrix} A_c & 0 \\ -C_c^T C_c & I \end{bmatrix}$$

getting $\begin{bmatrix} B_c B_c^T & A_c \\ A_c^T & -C_c^T C_c \end{bmatrix}$ as the Schur complement of the quasidefinite

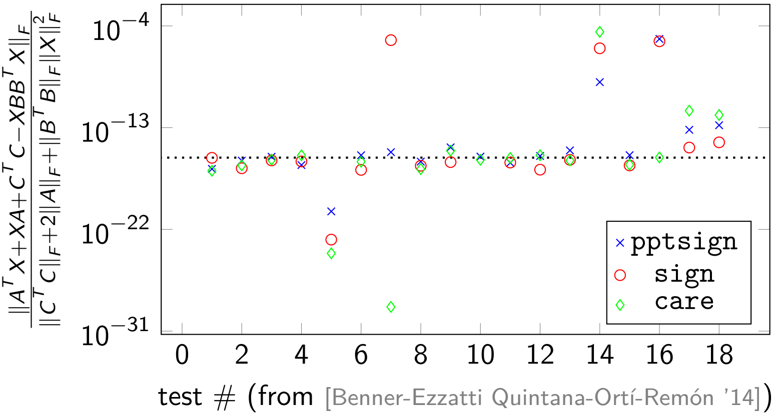
$$\left[\begin{array}{cc|cc} BB^T & 0 & A - I & -\sqrt{2}I \\ 0 & 0 & \sqrt{2}I & I \\ \hline A^T - I & \sqrt{2}I & -C^T C & 0 \\ -\sqrt{2}I & I & 0 & 0 \end{array}\right].$$

PPTs $\implies$ Cayley transforms $\implies$ Riccati solutions

Algorithm

1. Input: A, B, C.
2. Run sign iteration on $H_0 = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix}$ via PPTs, getting
$$H_\infty = \operatorname{sign}(H_0) = \begin{bmatrix} A_s & B_s B_s^T \\ C_s^T C_s & -A^T \end{bmatrix}.$$
3. Compute Cayley transform
$$(H_\infty - I)^{-1}(H_\infty + I) = \begin{bmatrix} I & B_c B_c^T \\ 0 & A_c^T \end{bmatrix}^{-1} \begin{bmatrix} A_c & 0 \\ -C_c^T C_c & I \end{bmatrix}$$ via PPTs.
4. Then, $X = C_c^T C_c$, $Y = B_c B_c^T$ solve the two CAREs.

Some preliminary experiments



- Some improvement on `sign`.
- Returns factored iterates natively.
- Still some work to do!

PPTs $\implies$ sign iteration

Possible solution: More PPTs!

One step of the sign iteration

$$H_0 = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix} \mapsto \frac{1}{2}(H_0 + H_0^{-1}) = \begin{bmatrix} A_1 & B_1 B_1^T \\ C_1^T C_1 & -A_1^T \end{bmatrix}$$

can be interpreted as a Schur complement

$$\begin{bmatrix} B_1 B_1^T & A_1 \\ A_1^T & -C_1^T C_1 \end{bmatrix} = \left[\begin{array}{cc|cc} BB^T & 0 & A & I \\ 0 & BB^T & -I & A \\ \hline A^T & -I & -C^T C & 0 \\ I & A^T & 0 & -C^T C \end{array}\right].$$

And so can various other operations; e.g., a step of structured doubling algorithm [Chu-Fan-Lin-Wang '04].

Conclusions

- What we did: factored PPTs $\implies$ quasidefinite inverses $\implies$ matrix sign $\implies$ Riccati solutions.
- PPTs are an unusual but elegant tool "from another planet" for linear algebra.
- Ask yourself: can I write this as a Schur complement / PPT?
- Quasi-definite / saddle-point matrices fit naturally in this framework.
- On the TO-DO list: run these algorithms not on $H = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix}$ and its blocks, but on its version with $\max_{ij} |M|_{ij} \leq \sqrt{2}$ (as in [Mehrmann P '12] for the structured doubling algorithm).
- Co-authors: Volker Mehrmann, Nataša Strabić, Peter Benner.

Conclusions

- What we did: factored PPTs $\implies$ quasidefinite inverses $\implies$ matrix sign $\implies$ Riccati solutions.
- PPTs are an unusual but elegant tool “from another planet” for linear algebra.
- Ask yourself: can I write this as a Schur complement / PPT?
- Quasi-definite / saddle-point matrices fit naturally in this framework.
- On the TO-DO list: run these algorithms not on $H = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix}$ and its blocks, but on its version with $\max_{ij} |M|_{ij} \leq \sqrt{2}$ (as in [Mehrmann P '12] for the structured doubling algorithm).
- Co-authors: Volker Mehrmann, Nataša Strabić, Peter Benner.

. . . and many thanks to $\begin{bmatrix} \mathrm{I} & \mathrm{L} \\ \mathrm{A} & \mathrm{S} \end{bmatrix}$