A new low-rank solver for algebraic Riccati equations based on the matrix sign function and principal pivot transforms

Peter Benner
joint work with Federico Poloni (Università di Pisa)

ILAS 2023
25th Conference of the International Linear Algebra Society
MSI04 “Matrix equations”
Madrid, June 12–16, 2023

Outline

Algebraic Riccati equation (ARE)

For $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

Algebraic Riccati equation (ARE)

For $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

Many applications:

- model reduction of (unstable) linear time-invariant (LTI) systems,
- linear-quadratic optimal control problems for LTI systems,
- H_∞-control, ...

Algebraic Riccati equation (ARE)

For $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

Many applications:

- model reduction of (unstable) linear time-invariant (LTI) systems,
- linear-quadratic optimal control problems for LTI systems,
- H_∞-control, ...

Typical situation in model reduction and control:

- G, W low-rank with $G, W \in \{BB^T, C^T C\}$, where $B \in \mathbb{R}^{n \times m}$, $m \ll n$, and $C \in \mathbb{R}^{p \times n}$, $p \ll n$.
- **Want:** solution with $X = X^T \geq 0$ (and $\Lambda(A - GX) \subset \mathbb{C}^-$), notation: $X_\geq$.

Algebraic Riccati equation (ARE)

For $A, G = G^T, W = W^T \in \mathbb{R}^{n\times n}$ given and $X \in \mathbb{R}^{n\times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

Many applications:

- model reduction of (unstable) linear time-invariant (LTI) systems,
- linear-quadratic optimal control problems for LTI systems,
- H_∞-control, ...

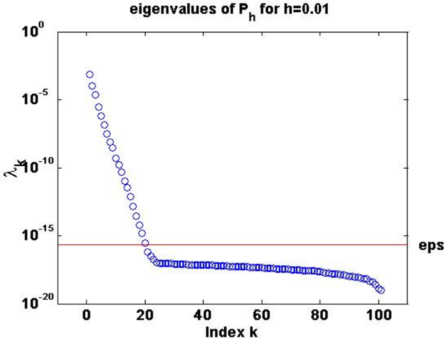
Typical situation in model reduction and control:

- G, W low-rank with $G, W \in \{BB^T, C^T C\}$, where $B \in \mathbb{R}^{n\times m}$, $m \ll n$, and $C \in \mathbb{R}^{p\times n}$, $p \ll n$.
- **Want:** solution with $X = X^T \geq 0$ (and $\Lambda\,(A - GX) \subset \mathbb{C}^-$), notation: $X_\geq$.
- $n = 10^3 - 10^6$
  $\Longrightarrow$ X has $10^6 - 10^{12}$ unknowns
  $\Longrightarrow$ as X is dense in general, we face a storage problem!

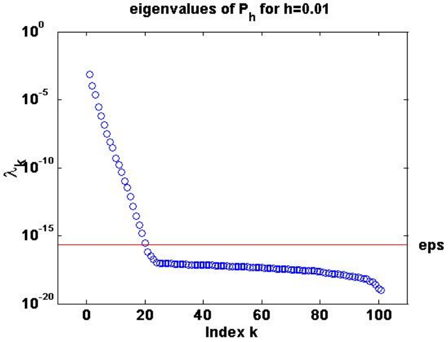
Consider spectrum of ARE solution.

Example:

- Linear 1D heat equation with point control,
- $\Omega = [0, 1]$,
- FEM discretization using linear B-splines,
- $h = 1/100 \Longrightarrow n = 101$.



Consider spectrum of ARE solution.



Example:

- Linear 1D heat equation with point control,
- $\Omega = [0, 1]$,
- FEM discretization using linear B-splines,
- $h = 1/100 \implies n = 101$.

Idea: $X = X^T \geq 0 \implies$

$$X = ZZ^T = \sum_{k=1}^{n} \lambda_k z_k z_k^T \approx \sum_{k=1}^{r} \lambda_k z_k z_k^T = \sum_{k=1}^{r} \left(\sqrt{\lambda_k} z_k\right) \left(\sqrt{\lambda_k} z_k\right)^T =: Z^{(r)}(Z^{(r)})^T.$$

$\implies$ **Goal:** compute $Z^{(r)} \in \mathbb{R}^{n \times r}$ directly w/o ever forming X!

$\exists$ many numerical methods to solve AREs.

Here: revisit the **matrix sign function method.**

$\exists$ many numerical methods to solve AREs.
Here: revisit the **matrix sign function method.**

Definition

For $Z \in \mathbb{R}^{n \times n}$ with $\Lambda(Z) \cap \imath\mathbb{R} = \emptyset$ and Jordan canonical form

$$Z = S \begin{bmatrix} J^+ & 0 \\ 0 & J^- \end{bmatrix} S^{-1}$$

the matrix sign function is

$$\operatorname{sign}(Z) := S \begin{bmatrix} I_k & 0 \\ 0 & -I_{n-k} \end{bmatrix} S^{-1}.$$

$\exists$ many numerical methods to solve AREs.
Here: revisit the **matrix sign function method.**

Definition

For $Z \in \mathbb{R}^{n\times n}$ with $\Lambda(Z) \cap \imath\mathbb{R} = \emptyset$ and Jordan canonical form

$$Z = S \begin{bmatrix} J^+ & 0 \\ 0 & J^- \end{bmatrix} S^{-1}$$

the matrix sign function is

$$\operatorname{sign}(Z) := S \begin{bmatrix} I_k & 0 \\ 0 & -I_{n-k} \end{bmatrix} S^{-1}.$$

Lemma

Let $T \in \mathbb{R}^{n\times n}$ be nonsingular and Z as before, then

$$\operatorname{sign}\left(TZT^{-1}\right) = T \operatorname{sign}(Z)\, T^{-1}.$$

$\exists$ many numerical methods to solve AREs.
Here: revisit the **matrix sign function method.**

Computation of $\operatorname{sign}(Z)$

$\operatorname{sign}(Z)$ is root of $I_n \Longrightarrow$ use Newton's method to compute it:

$$Z_0 \leftarrow Z, \qquad Z_{j+1} \leftarrow \frac{1}{2}\left(c_j Z_j + \frac{1}{c_j} Z_j^{-1}\right), \qquad j = 1, 2, \ldots$$

$\Longrightarrow \quad \operatorname{sign}(Z) = \lim_{j\to\infty} Z_j.$

$c_j > 0$ is scaling parameter for convergence acceleration and rounding error minimization, e.g.

$$c_j = \sqrt{\frac{\left\|Z_j^{-1}\right\|_F}{\|Z_j\|_F}},$$

based on "equilibrating" the norms of the two summands [Higham 1986].

Key observations:

Key observations:

1. Let $H = \begin{bmatrix} A & G \\ W & -A^T \end{bmatrix}$ be the Hamiltonian matrix associated to the ARE and $X_\geq$ the desired symmetric positive semidefinite solution. Then

$$H \begin{bmatrix} I_n \\ -X_\geq \end{bmatrix} = \begin{bmatrix} I_n \\ -X_\geq \end{bmatrix} (A - GX_\geq),$$

i.e., $X_\geq$ defines an n-dimensional invariant subspace of H corresponding to its left-half-plane eigenvalues.

Key observations:

1. Let $H = \begin{bmatrix} A & G \\ W & -A^T \end{bmatrix}$ be the Hamiltonian matrix associated to the ARE and $X_\geq$ the desired symmetric positive semidefinite solution. Then

$$H \begin{bmatrix} I_n \\ -X_\geq \end{bmatrix} = \begin{bmatrix} I_n \\ -X_\geq \end{bmatrix} (A - GX_\geq),$$

   i.e., $X_\geq$ defines an n-dimensional invariant subspace of H corresponding to its left-half-plane eigenvalues.

2. $\frac{1}{2}(I_{2n} + \operatorname{sign}(H))$ is a projector onto the H-invariant subspace corresponding to its left-half-plane eigenvalues

$$\Longrightarrow \qquad (I_{2n} + \operatorname{sign}(H)) \begin{bmatrix} I_n \\ -X_\geq \end{bmatrix} = 0.$$

Key observations:

1. Let $H = \begin{bmatrix} A & G \\ W & -A^T \end{bmatrix}$ be the Hamiltonian matrix associated to the ARE and $X_{\geq}$ the desired symmetric positive semidefinite solution. Then

$$H \begin{bmatrix} I_n \\ -X_{\geq} \end{bmatrix} = \begin{bmatrix} I_n \\ -X_{\geq} \end{bmatrix} (A - GX_{\geq}),$$

   i.e., $X_{\geq}$ defines an n-dimensional invariant subspace of H corresponding to its left-half-plane eigenvalues.

2. $\frac{1}{2}(I_{2n} + \text{sign}(H))$ is a projector onto the H-invariant subspace corresponding to its left-half-plane eigenvalues

$$\Longrightarrow \qquad (I_{2n} + \text{sign}(H)) \begin{bmatrix} I_n \\ -X_{\geq} \end{bmatrix} = 0.$$

3. Hence, $X_{\geq}$ is determined by overdetermined, but consistent linear system of equations once $\text{sign}(H)$ is known.

Main Goal

So far:

Main Goal

So far:

- Solution of AREs via $\operatorname{sign}(H)$, fully dense computations!

So far:

- Solution of AREs via $\operatorname{sign}(H)$, fully dense computations!
- Newton iteration for $\operatorname{sign}(H)$ preserves structure, as inversion and addition preserve Hamiltonian structure.

Main Goal

So far:

- Solution of AREs via $\mathrm{sign}(H)$, fully dense computations!
- Newton iteration for $\mathrm{sign}(H)$ preserves structure, as inversion and addition preserve Hamiltonian structure.
- But:
  1. off-diagonal blocks are not treated in low-rank format,
  2. $X_{\geq}$ cannot be determined in factored form directly from this.

So far:

- Solution of AREs via $\operatorname{sign}(H)$, fully dense computations!
- Newton iteration for $\operatorname{sign}(H)$ preserves structure, as inversion and addition preserve Hamiltonian structure.
- But:
  1. off-diagonal blocks are not treated in low-rank format,
  2. $X_{\geq}$ cannot be determined in factored form directly from this.

Goals

1. Keep the off-diagonal blocks in H in low-rank form — this would save a significant amount of memory, i.e., working with A, B, C directly would reduce memory requirements by a factor of $\sim 3-4$.

Main Goal

So far:

- Solution of AREs via $\operatorname{sign}(H)$, fully dense computations!
- Newton iteration for $\operatorname{sign}(H)$ preserves structure, as inversion and addition preserve Hamiltonian structure.
- But:
  1. off-diagonal blocks are not treated in low-rank format,
  2. $X_{\geq}$ cannot be determined in factored form directly from this.

Goals

1. Keep the off-diagonal blocks in H in low-rank form — this would save a significant amount of memory, i.e., working with A, B, C directly would reduce memory requirements by a factor of $\sim 3 - 4$.
2. Obtain $X_{\geq}$ in low-rank factored form directly.

Theorem (Kenney/Laub/Jonckheere 1989, B. 2019/22)

Let (A, B) be stabilizable, (A, C) be detectable, and define the Hamiltonian matrix

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

Theorem (Kenney/Laub/Jonckheere 1989, B. 2019/22)

Let (A, B) be stabilizable, (A, C) be detectable, and define the Hamiltonian matrix

$$\begin{bmatrix} A & -BB^T \\ -C^TC & -A^T \end{bmatrix}.$$

Then the unique stabilizing solution $X_{\geq}$ to the LQR Riccati equation exists and is symmetric positive semidefinite.

Theorem (Kenney/Laub/Jonckheere 1989, B. 2019/22)

Let (A, B) be stabilizable, (A, C) be detectable, and define the Hamiltonian matrix

$$\begin{bmatrix} A & -BB^T \\ -C^TC & -A^T \end{bmatrix}.$$

Then the unique stabilizing solution $X_\geq$ to the LQR Riccati equation exists and is symmetric positive semidefinite.

Hence, $A - BB^TX_s$ is stable, the closed-loop Lyapunov equations

$$\begin{aligned} (A - BB^TX_\geq)P + P(A - BB^TX_\geq)^T + BB^T &= 0, \\ (A - BB^TX_\geq)^TQ + Q(A - BB^TX_\geq) + C^TC &= 0, \end{aligned}$$

have unique solutions $P = P^T \geq 0$, $Q = Q^T \geq 0$, resp., and it holds

$$\operatorname{sign}(H) = \begin{bmatrix} -I + 2PX_\geq & -2P \\ 2X_\geq PX_\geq - 2X_\geq & I - 2X_\geq P \end{bmatrix}.$$

Theorem (Kenney/Laub/Jonckheere 1989, B. 2019/22)

Let (A, B) be stabilizable, (A, C) be detectable, and define the Hamiltonian matrix

$$\begin{bmatrix} A & -BB^T \\ -C^TC & -A^T \end{bmatrix}.$$

Then the unique stabilizing solution $X_\geq$ to the LQR Riccati equation exists and is symmetric positive semidefinite.

Hence, $A - BB^TX_s$ is stable, the closed-loop Lyapunov equations

$$\begin{aligned} (A - BB^TX_\geq)P + P(A - BB^TX_\geq)^T + BB^T &= 0, \\ (A - BB^TX_\geq)^TQ + Q(A - BB^TX_\geq) + C^TC &= 0, \end{aligned}$$

have unique solutions $P = P^T \geq 0$, $Q = Q^T \geq 0$, resp., and it holds

$$\operatorname{sign}(H) = \begin{bmatrix} -I + 2PX_\geq & -2P \\ 2X_\geq PX_\geq - 2X_\geq & I - 2X_\geq P \end{bmatrix}.$$

Hence, P (and by duality, Q) can be obtained from $\operatorname{sign}(H)$ directly, without solving the AREs at all, and in factored form if sign iterates preserve the off-diagonal low-rank structure!

Quasi-semidefinite Sign Function Iteration

Re-write the sign function iteration for $H = \begin{bmatrix} A & BB^T \\ C^TC & -A^T \end{bmatrix}$ in "symmetrized" form:

$$\frac{1}{2}\left(H + H^{-1}\right) = \frac{1}{2}\left(H + (J^TJH)^{-1}\right) = \frac{1}{2}\left(HJ^T + (JH)^{-1}\right)J =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right)J,$$

where

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} C^TC & -A^T \\ -A & -BB^T \end{bmatrix}, \qquad \tilde{M} = \begin{bmatrix} BB^T & -A \\ -A^T & -C^TC \end{bmatrix}.$$

Re-write the sign function iteration for $H = \begin{bmatrix} A & BB^T \\ C^TC & -A^T \end{bmatrix}$ in "symmetrized" form:

$$\frac{1}{2}\left(H + H^{-1}\right) = \frac{1}{2}\left(H + (J^TJH)^{-1}\right) = \frac{1}{2}\left(HJ^T + (JH)^{-1}\right)J =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right)J,$$

where

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} C^TC & -A^T \\ -A & -BB^T \end{bmatrix}, \qquad \tilde{M} = \begin{bmatrix} BB^T & -A \\ -A^T & -C^TC \end{bmatrix}.$$

Important observation: $M, \tilde{M}$ are **symmetric quasi-semidefinite** (SQSD).

Quasi-semidefinite Sign Function Iteration

Re-write the sign function iteration for $H = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix}$ in "symmetrized" form:

$$\frac{1}{2}\left(H + H^{-1}\right) = \frac{1}{2}\left(H + (J^T J H)^{-1}\right) = \frac{1}{2}\left(H J^T + (JH)^{-1}\right) J =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right) J,$$

where

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} C^T C & -A^T \\ -A & -BB^T \end{bmatrix}, \qquad \tilde{M} = \begin{bmatrix} BB^T & -A \\ -A^T & -C^T C \end{bmatrix}.$$

Important observation: $M, \tilde{M}$ are **symmetric quasi-semidefinite** (SQSD).

Here: inversion of SQSD matrices using **principal pivot transforms**

Quasi-semidefinite Sign Function Iteration

Re-write the sign function iteration for $H = \begin{bmatrix} A & BB^T \\ C^TC & -A^T \end{bmatrix}$ in "symmetrized" form:

$$\frac{1}{2}\left(H + H^{-1}\right) = \frac{1}{2}\left(H + (J^TJH)^{-1}\right) = \frac{1}{2}\left(HJ^T + (JH)^{-1}\right)J =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right)J,$$

where

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} C^TC & -A^T \\ -A & -BB^T \end{bmatrix}, \qquad \tilde{M} = \begin{bmatrix} BB^T & -A \\ -A^T & -C^TC \end{bmatrix}.$$

Important observation: $M, \tilde{M}$ are **symmetric quasi-semidefinite** (SQSD).

Here: inversion of SQSD matrices using **principal pivot transforms**

1. is numerically more robust than standard inversion of symmetric matrices,

Quasi-semidefinite Sign Function Iteration

Re-write the sign function iteration for $H = \begin{bmatrix} A & BB^T \\ C^TC & -A^T \end{bmatrix}$ in "symmetrized" form:

$$\frac{1}{2}\left(H + H^{-1}\right) = \frac{1}{2}\left(H + (J^TJH)^{-1}\right) = \frac{1}{2}\left(HJ^T + (JH)^{-1}\right)J =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right)J,$$

where

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} C^TC & -A^T \\ -A & -BB^T \end{bmatrix}, \qquad \tilde{M} = \begin{bmatrix} BB^T & -A \\ -A^T & -C^TC \end{bmatrix}.$$

Important observation: $M, \tilde{M}$ are **symmetric quasi-semidefinite** (SQSD).

Here: inversion of SQSD matrices using **principal pivot transforms**

1. is numerically more robust than standard inversion of symmetric matrices,
2. allows to work directly with A, B, C without ever forming $2n \times 2n$-matrices,

Quasi-semidefinite Sign Function Iteration

Re-write the sign function iteration for $H = \begin{bmatrix} A & BB^T \\ C^TC & -A^T \end{bmatrix}$ in "symmetrized" form:

$$\frac{1}{2}\left(H + H^{-1}\right) = \frac{1}{2}\left(H + (J^TJH)^{-1}\right) = \frac{1}{2}\left(HJ^T + (JH)^{-1}\right)J =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right)J,$$

where

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} C^TC & -A^T \\ -A & -BB^T \end{bmatrix}, \qquad \tilde{M} = \begin{bmatrix} BB^T & -A \\ -A^T & -C^TC \end{bmatrix}.$$

Important observation: $M, \tilde{M}$ are **symmetric quasi-semidefinite** (SQSD).

Here: inversion of SQSD matrices using **principal pivot transforms**

1. is numerically more robust than standard inversion of symmetric matrices,
2. allows to work directly with A, B, C without ever forming $2n \times 2n$-matrices,
3. yields a sign function iteration for AREs using A, B, C without ever forming $2n \times 2n$-matrices!

Definition (Principal Pivot Transform)

Let M be symmetric and invertible, and set $M^{(0)} := M, \quad W^{(0)} = \begin{bmatrix} M^{(0)} & -I_n \end{bmatrix}$.

Definition (Principal Pivot Transform)

Let M be symmetric and invertible, and set $M^{(0)} := M, \quad W^{(0)} = \begin{bmatrix} M^{(0)} & -I_n \end{bmatrix}$.

Select a $u \times u$ pivot block $M_{11}^{(0)}$ $(u \in \{1, 2\})$ and permute $W^{(0)}$ accordingly, call the result again $W^{(0)}$.

Definition (Principal Pivot Transform)

Let M be symmetric and invertible, and set $M^{(0)} := M, \quad W^{(0)} = \begin{bmatrix} M^{(0)} & -I_n \end{bmatrix}$.

Select a $u \times u$ pivot block $M_{11}^{(0)}$ $(u \in \{1,2\})$ and permute $W^{(0)}$ accordingly, call the result again $W^{(0)}$.

Define $K_0 = \begin{bmatrix} M_{11}^{(0)} & 0 \\ M_{21}^{(0)} & I_{n-u} \end{bmatrix}$ and compute

$$W^{(1)} := K_0^{-1} W^{(0)} = \begin{bmatrix} I_u & M_{12}^{(1)} & M_{11}^{(1)} & 0 \\ 0 & M_{22}^{(1)} & M_{21}^{(1)} & -I_{n-u} \end{bmatrix}.$$

Definition (Principal Pivot Transform)

Let M be symmetric and invertible, and set $M^{(0)} := M, \quad W^{(0)} = \begin{bmatrix} M^{(0)} & -I_n \end{bmatrix}$.

Select a $u \times u$ pivot block $M_{11}^{(0)}$ $(u \in \{1, 2\})$ and permute $W^{(0)}$ accordingly, call the result again $W^{(0)}$.

Define $K_0 = \begin{bmatrix} M_{11}^{(0)} & 0 \\ M_{21}^{(0)} & I_{n-u} \end{bmatrix}$ and compute

$$W^{(1)} := K_0^{-1} W^{(0)} = \begin{bmatrix} I_u & M_{12}^{(1)} & M_{11}^{(1)} & 0 \\ 0 & M_{22}^{(1)} & M_{21}^{(1)} & -I_{n-u} \end{bmatrix}.$$

Then

$$M^{(1)} := \begin{bmatrix} M_{11}^{(1)} & M_{12}^{(1)} \\ M_{21}^{(1)} & M_{22}^{(1)} \end{bmatrix} := \begin{bmatrix} -M_{11}^{-1} & M_{11}^{-1} M_{12} \\ M_{21} M_{11}^{-1} & M_{22} - M_{21} M_{11}^{-1} M_{12} \end{bmatrix}.$$

Definition (Principal Pivot Transform)

Let M be symmetric and invertible, and set $M^{(0)} := M, \quad W^{(0)} = \begin{bmatrix} M^{(0)} & -I_n \end{bmatrix}$.

Select a $u \times u$ pivot block $M_{11}^{(0)}$ ($u \in \{1, 2\}$) and permute $W^{(0)}$ accordingly, call the result again $W^{(0)}$.

Define $K_0 = \begin{bmatrix} M_{11}^{(0)} & 0 \\ M_{21}^{(0)} & I_{n-u} \end{bmatrix}$ and compute

$$W^{(1)} := K_0^{-1} W^{(0)} = \begin{bmatrix} I_u & M_{12}^{(1)} & M_{11}^{(1)} & 0 \\ 0 & M_{22}^{(1)} & M_{21}^{(1)} & -I_{n-u} \end{bmatrix}.$$

Then

$$M^{(1)} := \begin{bmatrix} M_{11}^{(1)} & M_{12}^{(1)} \\ M_{21}^{(1)} & M_{22}^{(1)} \end{bmatrix} := \begin{bmatrix} -M_{11}^{-1} & M_{11}^{-1} M_{12} \\ M_{21} M_{11}^{-1} & M_{22} - M_{21} M_{11}^{-1} M_{12} \end{bmatrix}.$$

The mapping $M^{(0)} \to M^{(1)}$ is called **principal pivot transform (PPT)**.

Definition (Principal Pivot Transform)

Let M be symmetric and invertible, and set $M^{(0)} := M, \quad W^{(0)} = \begin{bmatrix} M^{(0)} & -I_n \end{bmatrix}$.

Select a $u \times u$ pivot block $M_{11}^{(0)}$ ($u \in \{1, 2\}$) and permute $W^{(0)}$ accordingly, call the result again $W^{(0)}$.

Define $K_0 = \begin{bmatrix} M_{11}^{(0)} & 0 \\ M_{21}^{(0)} & I_{n-u} \end{bmatrix}$ and compute

$$W^{(1)} := K_0^{-1} W^{(0)} = \begin{bmatrix} I_u & M_{12}^{(1)} & M_{11}^{(1)} & 0 \\ 0 & M_{22}^{(1)} & M_{21}^{(1)} & -I_{n-u} \end{bmatrix}.$$

Then

$$M^{(1)} := \begin{bmatrix} M_{11}^{(1)} & M_{12}^{(1)} \\ M_{21}^{(1)} & M_{22}^{(1)} \end{bmatrix} := \begin{bmatrix} -M_{11}^{-1} & M_{11}^{-1} M_{12} \\ M_{21} M_{11}^{-1} & M_{22} - M_{21} M_{11}^{-1} M_{12} \end{bmatrix}.$$

The mapping $M^{(0)} \to M^{(1)}$ is called **principal pivot transform (PPT)**.

Repeating this m times with pivots of size u_k, so that $u_0 + \cdots + u_{m-1} = n$, yields $M^{(m)} = -M^{-1}$, i.e., a **Gauß-Jordan-type inversion** procedure for symmetric matrices.

Inversion of Symmetric Matrices

Most software packages compute inverses of symmetric matrices M using LDL^T factroization with Bunch-Kaufman (diagonal, partial) or Bunch-Parlett (complete) pivoting, e.g., xSYTRI from LAPACK and the MATLAB function `inv` based on this. SQSD structure is usually ignored, but turns out to be beneficial!

Most software packages compute inverses of symmetric matrices M using LDL^T factroization with Bunch-Kaufman (diagonal, partial) or Bunch-Parlett (complete) pivoting, e.g., xSYTRI from LAPACK and the MATLAB function `inv` based on this. SQSD structure is usually ignored, but turns out to be beneficial!

Theorem (Bunch–Parlett)

Let $LDL^T = \Pi M \Pi^T$ be the LDL^T factorization with Bunch–Parlett pivoting of a symmetric matrix M, with pivoting threshold $\tau = \frac{1+\sqrt{17}}{8} \approx 0.64$. Then,

$$\|D\|_{\max} \leq (2.57)^{n-1} \|M\|_{\max}, \quad \text{and} \quad \|L\|_{\max} \leq 2.78.$$

Here: a scalar pivot is chosen if $\max_{k=1,\ldots,n} |M[k,k]| \geq \tau \max_{i\neq j} |M[i,j]|$, if such k exists; otherwise maximum 2×2 pivot is chosen.

Inversion of Symmetric Matrices

Most software packages compute inverses of symmetric matrices M using LDL^T factroization with Bunch-Kaufman (diagonal, partial) or Bunch-Parlett (complete) pivoting, e.g., xSYTRI from LAPACK and the MATLAB function `inv` based on this. SQSD structure is usually ignored, but turns out to be beneficial!

Theorem (Bunch–Parlett)

Let $LDL^T = \Pi M \Pi^T$ be the LDL^T factorization with Bunch–Parlett pivoting of a symmetric matrix M, with pivoting threshold $\tau = \frac{1+\sqrt{17}}{8} \approx 0.64$. Then,

$$\|D\|_{\max} \leq (2.57)^{n-1} \|M\|_{\max}, \quad \text{and} \quad \|L\|_{\max} \leq 2.78.$$

Here: a scalar pivot is chosen if $\max_{k=1,\ldots,n} |M[k,k]| \geq \tau \max_{i \neq j} |M[i,j]|$, if such k exists; otherwise maximum 2×2 pivot is chosen.

Worst-case element growth can be slightly improved for SQSD matrices:

Theorem (B./Poloni 2019)

Let $LDL^T = \Pi M \Pi^T$ be the LDL^T factorization with Bunch–Parlett pivoting of a SQSD matrix M, with pivoting threshold $\tau = 1$. Then,

1. $\|D\|_{\max} \leq 2^{n-1} \|M\|_{\max}$, *and* $\|L\|_{\max} \leq 2$.
2. $\left\| |D| \left| D^{-1} \right| \right\|_{\max} \leq 2$, *and* $\left\| |D| \left| D^{-1} \right| \right\| \leq 3$.

Theorem (Backward stability of symmetric GJE)

Let $\hat{X}$ be the approximation of $X = -M^{-1}$ computed by the PPT-based symmetric Gauss–Jordan elimination algorithm. Then, each column $\hat{x}_j = \hat{X} e_j$ satisfies

$$-e_j = (M + \Delta_j)\hat{x}_j, \quad |\Delta_j| \leq |M| \left| L^{-T} \right| \left| L^T \right| \varepsilon_n,$$

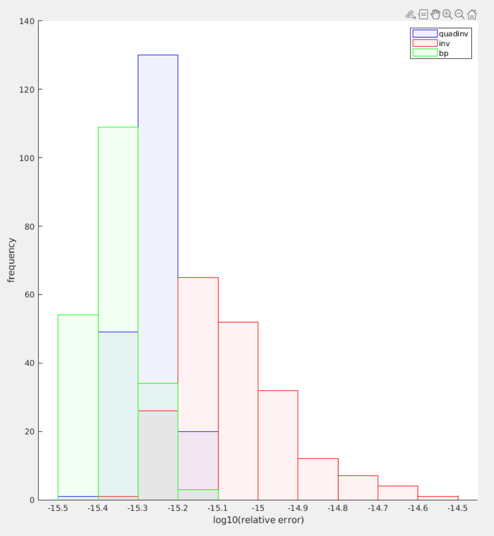
where $\varepsilon_n := \frac{cn\mathbf{u}}{1 - cn\mathbf{u}}$ with a constant c independent of n.

Inversion of 200 random 200×200 SQD matrices with

- MATLAB `inv`, based on DSYTRI from LAPACK
- Bunch-Parlett with complete pivoting,
- structured inversion using PPTs.

Inversion of 200 random 200×200 SQD matrices with

- MATLAB `inv`, based on DSYTRI from LAPACK
- Bunch-Parlett with complete pivoting,
- structured inversion using PPTs.



Factored Form of Sign Function Iteration

Recall: $M_+ =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right) J$ with $M, \tilde{M}$ SQSD.

Factored Form of Sign Function Iteration

Recall: $M_+ =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right) J$ with $M, \tilde{M}$ SQSD.

- For inversion of $M = \begin{bmatrix} C^T C & -A^T \\ -A & -BB^T \end{bmatrix}$, use symmetric GJE based on PPT.

Factored Form of Sign Function Iteration

Recall: $M_+ =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right) J$ with $M, \tilde{M}$ SQSD.

- For inversion of $M = \begin{bmatrix} C^T C & -A^T \\ -A & -BB^T \end{bmatrix}$, use symmetric GJE based on PPT.
- The inverse of a SQSD matrix is again SQSD, i.e.,

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

Factored Form of Sign Function Iteration

Recall: $M_+ =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right) J$ with $M, \tilde{M}$ SQSD.

- For inversion of $M = \begin{bmatrix} C^TC & -A^T \\ -A & -BB^T \end{bmatrix}$, use symmetric GJE based on PPT.
- The inverse of a SQSD matrix is again SQSD, i.e.,

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

- Inversion can be implemented using A, B, C only using again PPT-variant applied to generator matrix

$$\mathfrak{G} = \begin{bmatrix} B & A \\ * & C \end{bmatrix},$$

i.e., compute $\mathfrak{X} = \begin{bmatrix} \hat{B} & \hat{A} \\ * & \hat{C} \end{bmatrix}$ representing M^{-1} using A, B, C only without ever forming M [Poloni/Strabić 2016]!

Factored Form of Sign Function Iteration

Recall: $M_+ =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right) J$ with $M, \tilde{M}$ SQSD.

- For inversion of $M = \begin{bmatrix} C^TC & -A^T \\ -A & -BB^T \end{bmatrix}$, use symmetric GJE based on PPT.
- The inverse of a SQSD matrix is again SQSD, i.e.,

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

- Inversion can be implemented using A, B, C only using again PPT-variant applied to generator matrix

$$\mathfrak{G} = \begin{bmatrix} B & A \\ * & C \end{bmatrix},$$

  i.e., compute $\mathfrak{X} = \begin{bmatrix} \hat{B} & \hat{A} \\ * & \hat{C} \end{bmatrix}$ representing M^{-1} using A, B, C only without ever forming M [Poloni/Strabić 2016]!
- Update $M \to M_+$ can then be performed also on generators (potentially using rank truncation for offline blocks).

Factored Form of Sign Function Iteration

Recall: $M_+ =: \frac{1}{2}\left(\tilde{M} + M^{-1}\right) J$ with $M, \tilde{M}$ SQSD.

- For inversion of $M = \begin{bmatrix} C^T C & -A^T \\ -A & -BB^T \end{bmatrix}$, use symmetric GJE based on PPT.
- The inverse of a SQSD matrix is again SQSD, i.e.,

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

- Inversion can be implemented using A, B, C only using again PPT-variant applied to generator matrix

$$\mathfrak{G} = \begin{bmatrix} B & A \\ * & C \end{bmatrix},$$

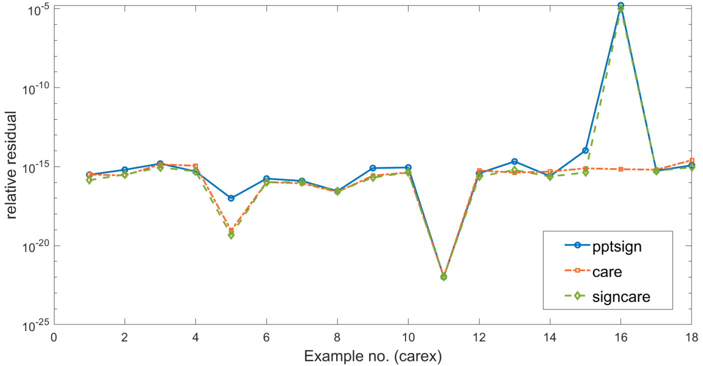
  i.e., compute $\mathfrak{X} = \begin{bmatrix} \hat{B} & \hat{A} \\ * & \hat{C} \end{bmatrix}$ representing M^{-1} using A, B, C only without ever forming M [Poloni/Strabić 2016]!
- Update $M \rightarrow M_+$ can then be performed also on generators (potentially using rank truncation for offline blocks).

$\rightsquigarrow$ new version of sign function iteration working directly on generators!

Test the new sign function iteration for AREs based on PPTs (pptsign)

- vs. MATLAB function care (based on Schur vector method) and classical sign function method (signcare) from MORLab [B./Werner 2006–2023],
- using 18 examples from carex benchmark collection [B./Laub/Mehrmann 1995].

Measure accuracy by $\frac{\left\|\mathcal{R}(\tilde{X})\right\|_F}{\left\|C^TC\right\|_F + 2\|A\|_F\left\|\tilde{X}\right\|_F + \left\|BB^T\right\|_F\left\|\tilde{X}^2\right\|_F}$.



Conclusions

- Symmetric quasi-semidefinite matrices can be inverted using PPT-based Gauß-Jordan type elimination in a structure-preserving and numerically robust way.
- Sign function iteration for AREs can be reformulated in terms of SQSD matrix inversions and summations, allowing to work with generator matrices (A, B, C) only, without ever forming $2n \times 2n$ matrices.
- Leads to much lower storage requirements and potentially to faster algorithms (fewer flops).
- Application: cloed-loop balanced truncation without ever solving AREs.
- Future work: sophisticated implementation to really test performance.

P. Benner.
An alternative algorithm for unstable balanced truncation.
MATHMOD 2022 Discussion Contributions, ARGESIM Report 17, pp. 71–72, 2022.
DOI: 10.11128/arep.17.a17178

P. Benner, P. Ezzatti, E.S. Quintana-Ortí, A. Remón.
A factored variant of the Newton iteration for the solution of algebraic Riccati equations via the matrix sign function.
Numerical Algorithms, 66:363—377, 2014.

P. Benner, M. Köhler, and J. Saak.
Matrix equations, sparse solvers: M-M.E.S.S.-2.0.1 – philosophy, features and application for (parametric) model order reduction.
In Model Reduction of Complex Dynamical Systems, P. Benner, T. Breiten, H. Faßbender, M. Hinze, T. Stykel, and R. Zimmermann, eds., ISNM 171, Birkhäuser, Cham, pp. 369–392, 2021.

P. Benner and S. W. R. Werner.
MORLAB – A model order reduction framework in MATLAB and Octave.
In Mathematical Software – ICMS 2020, A. M. Bigatti, J. Carette, J. H. Davenport, M. Joswig, and T. de Wolff, eds., LNCS 12097, Springer International Publishing, Cham, pp. 432–441, 2020.

C. Kenney, A. J. Laub, and E. A. Jonckheere.
Positive and negative solutions of dual Riccati equations by matrix sign function iteration.
Systems Control Letters, 13:109–116, 1989.

F. Poloni and N. Strabić.
Principal pivot transforms of quasidefinite matrices and semidefinite Lagrangian subspaces.
Electronic Journal of Linear Algebra, 31:200–231, 2016.

Commercial: METT-X

METT X
10th Workshop on Matrix Equations and Tensor Techniques

September 13–15, 2023
RWTH Aachen University (main building)

https://www.igpm.rwth-aachen.de/workshop/mett2023

Special Issue of ETNA (Electronic Transactions on Numerical Analysis),
open for participants only!
Fully (diamond) Open Access without OA charges!