

Algorithms for quadratic matrix equations in probability

With an eye to similarities with Control Theory

Federico Poloni

Scuola Normale Superiore, Pisa — Now @TU Berlin

TU Berlin, 29 April 2010

Everyone has problems

I will be at TU till June, 4

Looking forward to working & discussing research problems (either yours or mine) close to my research area while I'm here

My problems

- 1 Matrix equations in probability
- 2 Matrix geometric means

This talk will be about #1

... but first a very short mention to #2

What I won't talk about

Problem A proper generalization of the **geometric mean** $\sqrt[k]{a_1 a_2 \dots a_k}$ to positive definite (PD) matrices

Applications Averaging the results of physical experiments which return PD (or maybe even Hamiltonian, symplectic) matrices

Connections to the **Riemannian geometry** of PD matrices

It is tricky how to define properly a geometric mean of $n \geq 3$ matrices satisfying some intuitive properties (permutation invariance, monotonicity)

Problems

- No unique solution: which one is to prefer?
- How to compute them efficiently?
- Find new easier-to-compute means

[Ando, Li, Mathias 2004] [Moakher 2005] [Bini, Meini, P. 2010] + others

What I'll talk about

In applied probability, several analogous to the matrix equations you are studying at TU:

Examples

Nonsymmetric algebraic Riccati equation

$$XCX - AX - XD + B = 0 \quad (\text{NARE})$$

Unilateral quadratic equation

$$PX^2 + QX + R = 0 \quad (\text{UQME})$$

(\Leftrightarrow quadratic eigenvalue problems)

Solutions \Leftrightarrow **invariant subspaces** of suitable structured matrices/pencils.

What changes — eigenvalues

- No symmetry/structure \Rightarrow no eigenvalue symmetry, instead:

Eigenvalue splitting

NARE: **Hamiltonian** splitting

n eigval's in the right half-plane, n in the left

UQME: **symplectic** splitting

n eigval's inside the unit circle, n outside

- Looking for the solution corresponding to the eigval's in the right half-plane/inside the circle
- convergence depends on the **gap** between the closest pair of eigval's in opposite regions
- These eigval's are usually real (Perron-Frobenius theory)
- Eigval's on the border, but only in selected locations: 0, 1, roots of 1
Often we can **remove** them (we'll see later)
- Troubles: eigenvalues **very close** to the border (**critical cases**)

What changes — positivity

- New structure to preserve: **positivity** (entry-wise)
- Because of this, **iterative methods** are usually preferred to Schur form-based ones
- Schur-based exist, e.g. [Guo 2006], but more troubles with ill-conditioned cases
- Probabilists like algorithms with **probabilistic interpretations**
- They work with moderate sizes (for now), so we may use **dense linear algebra** – $O(n^3)$

Younger field, less understood than control theory

We could borrow some of your theory/tools

Nonsymmetric ARE

Nonsymmetric algebraic Riccati equation

$$XCX - AX - XD + B = 0 \quad (\text{NARE})$$

with **Hamiltonian splitting**

Possible algorithms:

- Fixed-point iterations — **slow** convergence
- Newton's method — **expensive**: requires solving a “nonsymmetric Lyapunov” at each step
- Structured Doubling Algorithm (SDA) — much better

SDA was born for control theory (DARE/CARE) [Anderson '78] [Chu, Fan, Lin 2005] but gained more success here

Outline of SDA

- 1 Construct $H = \begin{bmatrix} D & -C \\ B & -A \end{bmatrix}$ (eigval's have **Hamiltonian** splitting)
- 2 Cayley transform $S = \frac{H-\gamma I}{H+\gamma I}$ (**symplectic** splitting)
- 3 Factor $S = \begin{bmatrix} I & -G_0 \\ 0 & F_0 \end{bmatrix}^{-1} \begin{bmatrix} E_0 & 0 \\ -H_0 & I \end{bmatrix}$
- 4 Find E_1, F_1, G_1, H_1 of an analogous factorization of S^2
- 5 Repeat: $S^{2^k} = \begin{bmatrix} I & -G_k \\ 0 & F_k \end{bmatrix}^{-1} \begin{bmatrix} E_k & 0 \\ -H_k & I \end{bmatrix}$

You can also think in terms of pencils: $\lambda \begin{bmatrix} I & -G_k \\ 0 & F_k \end{bmatrix} - \begin{bmatrix} E_k & 0 \\ -H_k & I \end{bmatrix}$

SDA – formulas and convergence

SDA

$$S^{2^k} = \begin{bmatrix} I & -G_k \\ 0 & F_k \end{bmatrix}^{-1} \begin{bmatrix} E_k & 0 \\ -H_k & I \end{bmatrix} \quad \text{or} \quad \lambda \begin{bmatrix} I & -G_k \\ 0 & F_k \end{bmatrix} - \begin{bmatrix} E_k & 0 \\ -H_k & I \end{bmatrix}$$

$$E_{k+1} = E_k(I - G_k H_k)^{-1} E_k$$

$$F_{k+1} = F_k(I - H_k G_k)^{-1} F_k$$

$$G_{k+1} = G_k + E_k(I - G_k H_k)^{-1} H_k E_k$$

$$H_{k+1} = H_k + F_k(I - H_k G_k)^{-1} H_k F_k$$

$E_k \rightarrow 0 \sim$ eigval's in the unit circle converge to zero

$F_k \rightarrow 0 \sim$ eigval's outside the circle converge to eigval's at infinity

$G_k \rightarrow X_+$, $H_k \rightarrow X_-$ solutions associated with stable/unstable subspaces

we only invert the “tame” matrices $I - G_k H_k$, $I - H_k G_k$

SDA vs. matrix sign iteration

Why is squaring the “right thing”?

$$\begin{array}{ccc} H & \xrightarrow{\text{Sign iter}} & \frac{1}{2} (H + H^{-1}) \\ \downarrow \text{Cayley} & & \downarrow \text{Cayley} \\ S & \xrightarrow{\text{Squaring}} & S^2 \end{array}$$

SDA \Leftrightarrow matrix sign iteration — more robust implementation

Converges even with (multiple) eig'vals on the border (linear instead of quadratic convergence) [Guo, Lin + 4 others, 2009]

Good accuracy even in critical cases, when $(I - G_k H_k) \rightarrow$ singular. **Why?**

Also works when X is not square (n eigval's left, m right)

Questions

Questions

- Scaling the matrix sign iteration inside SDA
- SDA for Lur'e equations? Work in progress with Timo...
- Weaker applicability conditions
- Other choices of the factorization/pencil
- Satisfying error theory
- Are SDA and Newton's method related?

Unilateral quadratic equations

Unilateral quadratic equation

$$PX^2 + QX + R = 0 \quad (\text{UQME})$$

P and R might be (highly) singular

with **symplectic** splitting

Algorithms:

- No one attempted direct methods (linearization + Schur) as far as I know
- Fixed-point iterations — **slow** convergence
- Newton's method — **expensive**
- Cyclic Reduction (CR) — much better

CR was born for PDE [Golub, 70], rediscovered for these equations [Latouche, Ramaswami '93] [Bini, Meini '96]

Outline of CR

- 1 Construct an equation for X^2 :

$$0 = (PX^2 + QX + R)Q^{-1}(PX^2 - QX + R) = P_1(X^2)^2 + Q_1X^2 + R_1$$

- 2 Repeat

The pencil $\lambda^2 P_1 + \lambda Q_1 + R_1$ has the same eigvec's as $\lambda^2 P + \lambda Q + R$, but squared eigenvalues

CR – formulas and convergence

CR

$$P_k \lambda^2 + Q_k \lambda + R_k$$

$$P_{k+1} = P_k Q_k^{-1} R_k$$

$$Q_{k+1} = Q_k - P_k Q_k^{-1} R_k - R_k Q_k^{-1} P_k$$

$$R_{k+1} = R_k Q_k^{-1} R_k$$

$R_k \rightarrow 0 \sim$ eigval's in the unit circle converge to zero

$P_k \rightarrow 0 \sim$ eigval's outside the circle converge to eigval's at infinity
(some expressions) $\rightarrow X_{\pm}$, solutions associated with stable/unstable subspaces can be recovered

Converges even with (multiple) eig'vals on the border (linear instead of quadratic)

Good accuracy even in critical cases, when $Q_k \rightarrow$ singular

What's going on: SDA, CR and linearizations

Connections between them:

- Linearization

$$\lambda^2 P + \lambda Q + R \mapsto \lambda \begin{bmatrix} 0 & I \\ R & 0 \end{bmatrix} + \begin{bmatrix} P & 0 \\ Q & -I \end{bmatrix}$$

followed by (a variant of) SDA yields CR [Guo, Lin + others 2009]

- Quadraticization

$$\begin{bmatrix} I & -G_0 \\ 0 & F_0 \end{bmatrix} \lambda - \begin{bmatrix} E_0 & 0 \\ -H_0 & I \end{bmatrix} \mapsto \lambda^2 \begin{bmatrix} 0 & 0 \\ 0 & F \end{bmatrix} - \lambda \begin{bmatrix} I & -G \\ -H & I \end{bmatrix} + \begin{bmatrix} E & 0 \\ 0 & 0 \end{bmatrix}$$

(adds n eigval's at 0, n at infinity) followed by CR yields SDA [Bini, Meini, P. 2008]

Questions

The same as SDA!

Questions

- Scaling the matrix sign iteration inside CR
- Weaker applicability conditions
- Other choices of the linearization
- Satisfying error theory
- Are CR and Newton's method related?

The shift technique

How to deal with eigval's on the border?

Theorem

Let (λ, v) be an eigenpair of H ; for each μ, u

$$\tilde{H} = H + \mu v u^T$$

has the same eigval's of H , except λ becomes $\lambda + \mu$

Solutions to the matrix eqns change in predictable ways
(or in some cases do not change at all)

Eigvec's associated with 0 (or 1) are known or simple to compute —
shifting of 0 (1) eigval's is customary [He, Meini, Rhee 2001]

Shift for critical cases

Recently we tried to compute directly critical eigenspaces even for **eigval's close to the border** and shift them away [Iannazzo, P. submitted]

You can do most of the job working with **well-conditioned** eigenspaces instead of **ill-conditioned** eigenpairs

Mixed method: partly direct eig computation, partly iteration

Questions

- Better ways to compute the critical eigenspaces (Arnoldi variants?)
- Detailed error analysis – some parts are $O(\sqrt{\epsilon})$, some are $O(\epsilon)$

Lu's "simple equation"

A vector equation arising from a rank-structured NARE

Lu's equation

$$\begin{aligned}x &= (Py) .* x + \text{ones}(n, 1) \\y &= (Qx) .* y + \text{ones}(n, 1)\end{aligned} \quad x, y \in \mathbb{R}_+^n \text{ unknowns} \quad (\text{SE})$$

You can use **displacement rank-structured** linear algebra and perform iterative algorithms (SDA, Newton) in $O(n^2)$ ops/step [Bini, Iannazzo, P. 2008]

... or you can recover the solution explicitly after computing the eigenvalues [Mehrmann, Xu 2008]

But how do they compute the eigenvalues? Through a quadratically convergent $O(n^2)$ ops/step iteration

Question

Are these iterative and "direct" methods related?

Markovian binary trees

MBT equation

$$x = a + b(x, x)$$

$a, x \in \mathbb{R}_+^n$ and $b : \mathbb{R}_+^n \times \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ vector-valued **bilinear map**

$e := \text{ones}(n, 1)$ always a solution, but we want the **minimal nonnegative**

Algorithms:

- Fixed-point iterations — always slow
- Newton's method — slow convergence when $x_{\min} \approx e$

New algorithm: Perron vector-based iterations [Bini, Meini, P. submitted]

Outline of PV-based

- 1 Change of vars $y := e - x$ yields

$$y = (H_y)y, \quad H_y \in \mathbb{R}_+^{n \times n} \text{ depending linearly on } y$$

- 2 View as $y = \text{PV}(H_y)$, where $\text{PV}(\cdot)$ **Perron vector** (eigvec with $|\lambda| = \max$) with suitable normalization
- 3 Solve $y = \text{PV}(H_y)$ with fixed-point/Newton iteration

Convergence **does not get slower** when $x_{min} \approx e$

We have “deflated” the known solution $x = e$

Not “close to a double solution” anymore

Problems

- Convergence conditions (ok in all practical cases, but not globally)

A unifying framework

Why is this promising?

$$Mx = a + b(x, x)$$

$$XCX - AX - XD + B = 0 \quad (\text{NARE})$$

$$PX^2 + QX + R = 0 \quad (\text{UQME})$$

$$\begin{cases} Ix = (Py) .* x + \text{ones}(n, 1) \\ Iy = (Qx) .* y + \text{ones}(n, 1) \end{cases} \quad (\text{SE})$$

With a bit of $\text{vec}(\cdot)$, all the previous equations fall into this case

Although no more known ($x = e$) solution \rightarrow no PV-based algorithms

This insight leads to:

- Unified (and sometimes more general) proofs for Fixed-point + Newton – no **spectral theory** needed
- Unified (and sometimes faster) algorithms

Questions

Questions

- Adapt SDA/CR to this framework
- Adapt PV-based algorithms to NARE/UQME
- Include spectral theory in this framework
- Include shift

Thanks for your attention!