Exploiting displacement structure in the solution of a class of nonsymmetric algebraic Riccati equations

D. A. Bini B. Iannazzo B. Meini F. Poloni

Università di Pisa
Scuola Normale Superiore, Pisa

2nd intl. conference on matrix methods and operator equations
Moscow, 23 July 2007

Outline

Algebraic Riccati equations

Nonsymmetric algebraic Riccati equation (NARE)

$$\begin{aligned} XCX - AX - XE + B = 0 \\ A, B, C, E, X \in \mathbb{R}^{n \times n} \end{aligned} \qquad \text{(NARE)}$$

Recent interest in the literature e.g. [Guo–Laub '00, Lu '05, Guo–Higham '05, Bini–Iannazzo–Latouche–Meini '06]

Algebraic Riccati equations

Nonsymmetric algebraic Riccati equation (NARE)

$$XCX - AX - XE + B = 0 \qquad A, B, C, E, X \in \mathbb{R}^{n\times n} \tag{NARE}$$

Recent interest in the literature e.g. [Guo–Laub '00, Lu '05, Guo–Higham '05, Bini–Iannazzo–Latouche–Meini '06]

X solves (NARE) $\Leftrightarrow$ $\begin{bmatrix} E & -C \\ B & -A \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix} (E - CX)$

Solutions $\Leftrightarrow$ invariant subspaces of $\mathcal{H} := \begin{bmatrix} E & -C \\ B & -A \end{bmatrix}$

- Explicit calculation of the eigenvectors: numerical problems
- Iterative methods: cost $O(n^3)$/step, quadratic convergence

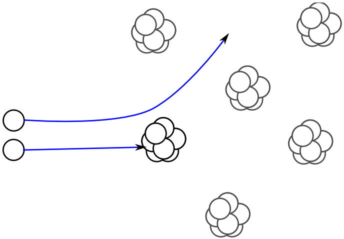
One-group neutron transport equation

$$\left\{(\mu+\alpha)\frac{\partial}{\partial x}+1\right\}\varphi(x,\mu)=\frac{c}{2}\int_{-1}^{1}\varphi(x,\omega)d\omega$$

$$\varphi(0,\mu)=f(\mu),\quad \mu>-\alpha,\quad |\mu|\leqslant 1,$$

$$\lim_{x\to\infty}\varphi(x,\mu)=0.$$

Propagation of neutrons through a slab of shielding material



One-group neutron transport equation

$$\left\{(\mu+\alpha)\frac{\partial}{\partial x}+1\right\}\varphi(x,\mu)=\frac{c}{2}\int_{-1}^{1}\varphi(x,\omega)d\omega$$

$$\Downarrow$$

Reduction to kernel + Gaussian quadrature $\int_0^1 f(x)dx \approx \sum w_i f(x_i)$

$$\Downarrow$$

The resulting equation

$$\Delta X + XD = (Xq+e)(e^T+q^TX) \qquad \text{(NT)}$$

D,Δ "positive" diagonals, $e,q>0$ vectors

(NT) is a NARE with rank structure:

$$A=\Delta-eq^T,\ B=ee^T,\ C=qq^T,\ E=D-qe^T$$

Algorithms for NARE

1. Newton's method [C.H. Guo–Laub '00]
2. Newton applied to Lu's iteration [Lu '05] — only for (NT)
3. Structured doubling algorithm [X.X. Guo–Lin–Xu '06]
4. Cyclic reduction [Ramaswami '99 and others]

All with cost $O(n^3)$/step, quadratic convergence

Our results

- Structured versions for (NT), with cost $O(n^2)$/step
- Shift technique [He–Meini–Rhee '01] in the structured algorithms
- Interesting connections: $1 = 2$, $3 \subseteq 4$
- New variants to 4

Riccati equations and M-matrices

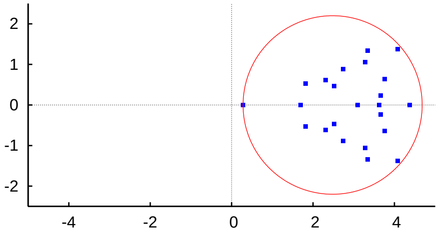
Classical hypothesis — includes case (NT)

$\mathcal{M} = \begin{bmatrix} E & -C \\ -B & A \end{bmatrix}$ is an M-matrix

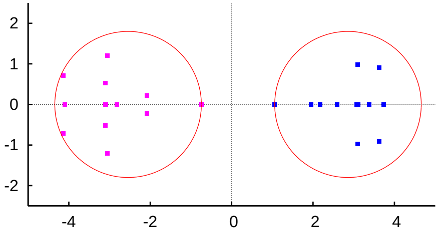
With this assumption,

- $\mathcal{H} = \begin{bmatrix} E & -C \\ B & -A \end{bmatrix}$ has:
  n eigenvalues in the positive half-plane $\Re(\lambda) > 0$
  n in the negative half-plane
  (eventually some on the border)
- Exists S minimal nonnegative solution
- $S \Leftrightarrow$ eigenvalues with $\Re(\lambda) > 0$
- The classical algorithms converge to S

Example



Eigenvalues of $\mathcal{M}$



Eigenvalues of $\mathcal{H}$

Cauchy-like matrices

Displacement operator

$$\nabla_{R,S}(X) := RX - XS$$

R, S diagonal matrices

Cauchy-like matrices: $\nabla_{R,S}(X)$ is low rank $\Leftrightarrow$

$$X_{ij} = \frac{u_i \cdot v_j}{R_{ii} - S_{jj}} \quad \text{whenever } R_{ii} \neq S_{jj}$$

u_i, v_j (*generators*) are $1 \times r$, $r \times 1$ vectors

Usually one requires $R_{ii} \neq S_{jj}$ for all i,j

Instead, we will also need the case $R = S$ (*Trummer-like*):
nothing is known about the main diagonal of X

The GKO algorithm

Solving linear systems with Cauchy-like matrices: GKO algorithm [Gohberg–Kailath–Olshevsky '95]

During Gaussian elimination,

$$M \longrightarrow \begin{bmatrix} c_{11} & c_{12} \\ 0 & C \end{bmatrix} \text{ with } C \text{ Cauchy-like}$$

Instead of updating the elements of C (cost: $O(n^3)$), update its generators (cost: $O(n^2)$)

Trummer-like case is similar:

- Update the diagonal of C as in the traditional Gaussian elimination $O(n^2)$
- Update the other elements as in GKO $O(n^2)$

Lu's algorithm

The resulting equation

$$\Delta X + XD = (Xq + e)(e^T + q^T X) \qquad \text{(NT)}$$

Let $u := Xq + e$, $v^T := e^T + q^T X$

$$(NT) \iff \nabla_{\Delta,-D}(X) = uv^T$$

X is Cauchy-like, and

$$\begin{cases} u = \nabla^{-1}_{\Delta,-D}(uv^T)q + e \\ v = \left(\nabla^{-1}_{\Delta,-D}(uv^T)\right)^T q + e \end{cases} \qquad \text{(LU)}$$

Let $w := \begin{bmatrix} u \\ v \end{bmatrix}$; (LU) is $F(w) = 0$, solve with Newton's method

$$w_{k+1} = w_k - (\nabla F_{w_k})^{-1} F(w_k)$$

The same in $O(n^2)$: ∇F is Trummer-like, use GKO

Newton's algorithm

NARE

$$XCX - AX - XE + B = 0$$

Newton's method applied directly to $R(X) = XCX - AX - XE + B$
The Jacobian is

$$\nabla R_X = I \otimes (A - XC) + (E - CX)^T \otimes I \qquad (n^2 \times n^2 \text{ matrix})$$

Or rather,

$$\nabla R_X(Y) = (A - XC)Y + Y(E - XC) \qquad \text{(SYL)}$$

We need ∇R_X^{-1}: solve (SYL), costs $O(n^3)$ (but slow)

Structured Newton for (NT)

$$\nabla R_X = (E - CX)^T \otimes I_n + I_n \otimes (A - XC) =$$

$$\underbrace{(D^T \otimes I_n + I_n \otimes \Delta)}_{\text{diagonal } n^2 \times n^2} - \underbrace{\begin{bmatrix}(e + X^T q \otimes I_n) & I_n \otimes (e + Xq)\end{bmatrix}}_{n^2 \times 2n} \underbrace{\begin{bmatrix} q^T \otimes I_n \\ I_n \otimes q^T \end{bmatrix}}_{2n \times n^2}$$

Sherman–Morrison–Woodbury formula

$$(\mathcal{D} - UV)^{-1} = \mathcal{D}^{-1} + \mathcal{D}^{-1} U (I_{2n} - V\mathcal{D}^{-1}U)^{-1} V\mathcal{D}^{-1}$$

We reduce to the inversion of $\mathcal{R} = I_{2n} - V\mathcal{D}^{-1}U$, $2n \times 2n$.

Moreover,

- $\mathcal{R}$ is Trummer-like (we can use GKO)
- $\mathcal{R}$ is well-conditioned

Relations between Newton and Lu

Lu: Invert ∇F_{w_k} Newton: Invert $\mathcal{R} = I_{2n} - V_k \mathcal{D}^{-1} U$

∇F_{w_k} and $\mathcal{R}$ have the same structure. A deeper connection?

Theorem

Let u_k, v_k be the iterates of Lu, starting from $u_{-1} = v_{-1} = 0$, and X_k be the iterates of Newton, starting from $X_0 = 0$. Then,

$$\begin{cases} u_k = X_k q + e \\ v_k = X_k^T q + e \end{cases} \qquad \forall k \geqslant 0$$

Interpretation Newton iterates are Cauchy-like:

$$\nabla_{\Delta,-D} X^{(k+1)} = u_{k+1} v_{k+1}^T - (u_{k+1} - u_k)(v_{k+1}^T - v_k^T)$$

Lu performs Newton's iteration working on the generators.

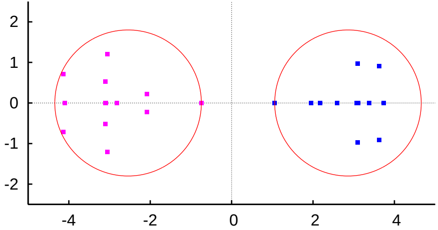
Structured doubling algorithm (SDA)

Structured doubling algorithm

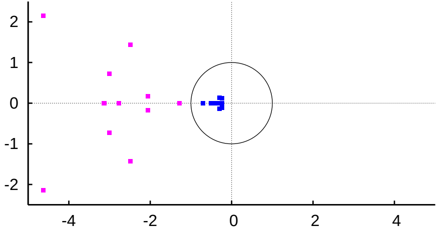
$$\begin{aligned} E_{k+1} &= E_k(I - G_k H_k)^{-1} E_k, \\ F_{k+1} &= F_k(I - H_k G_k)^{-1} F_k, \\ G_{k+1} &= G_k + E_k(I - G_k H_k)^{-1} G_k F_k, \\ H_{k+1} &= H_k + F_k(I - H_k G_k)^{-1} H_k E_k, \end{aligned} \tag{SDA}$$

1. Spectral transformation:
   $\mathcal{H} = \begin{bmatrix} E & -C \\ B & -A \end{bmatrix} \mapsto \mathcal{H}_\gamma := (\mathcal{H} + \gamma I)^{-1}(\mathcal{H} - \gamma I)$
2. Block UL factorization: $\mathcal{H}_\gamma = \mathcal{U}_0^{-1}\mathcal{L}_0$ con
   $$\mathcal{U} = \begin{bmatrix} I & -G_0 \\ 0 & F_0 \end{bmatrix}, \quad \mathcal{L} = \begin{bmatrix} E_0 & 0 \\ -H_0 & I \end{bmatrix}$$
3. Implicit update $\mathcal{H}_\gamma^{2^k} = \mathcal{U}_k^{-1}\mathcal{L}_k$

Example



Eigenvalues of $\mathcal{H}$



Eigenvalues of $\mathcal{H}_\gamma = (\mathcal{H} + \gamma I)^{-1}(\mathcal{H} - \gamma I)$

Fast SDA for (NT)

$$\mathcal{H}_\gamma^{2^k} = \begin{bmatrix} I & -G_k \\ 0 & F_k \end{bmatrix}^{-1} \begin{bmatrix} E_k & 0 \\ -H_k & I \end{bmatrix}$$

Cauchy-like structure

$$\begin{aligned}
DE_k - E_k D &= (q + G_k e)e^T E_k - E_k q(e^T + q^T H_k), \\
\Delta F_k - F_k \Delta &= (H_k q + e)q^T F_k - F_k e(e^T + q^T G_k), \\
DG_k + G_k \Delta &= (q + G_k e)(e^T + q^T G_k) - E_k q q^T F_k, \\
\Delta H_k + H_k D &= (H_k q + e)(e^T + q^T H_k) - E_k q q^T F_k,
\end{aligned}$$

Instead of updating E_k, F_k, G_k, H_k,

- update the above generators
- store and update the main diagonals of E_k, F_k

Cyclic reduction

NARE $\Leftrightarrow$ eigenvalue problem $\begin{bmatrix} E & -C \\ B & -A \end{bmatrix} u = \lambda u$

Multiply the second block column by λ:

$$\left(\begin{bmatrix} E & 0 \\ B & 0 \end{bmatrix} + \begin{bmatrix} -I & -C \\ 0 & -A \end{bmatrix} \lambda + \begin{bmatrix} 0 & 0 \\ 0 & -I \end{bmatrix} \lambda^2 \right) u = 0$$

yields a quadratic eigenvalue problem

Theorem

S solves the NARE $\Leftrightarrow$ $\begin{bmatrix} E - CS & 0 \\ S & 0 \end{bmatrix}$ solves the unilateral equation

$$\begin{bmatrix} E & 0 \\ B & 0 \end{bmatrix} + \begin{bmatrix} -I & -C \\ 0 & -A \end{bmatrix} X + \begin{bmatrix} 0 & 0 \\ 0 & -I \end{bmatrix} X^2 = 0 \qquad \text{(UNI)}$$

Cyclic reduction – the classical algorithm

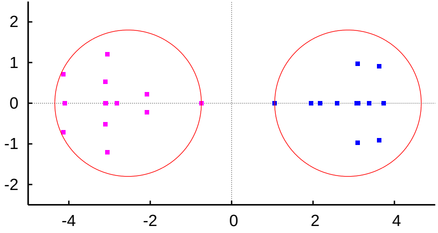
Cyclic reduction [Buzbee–Golub–Nielson, '69]

$$\begin{aligned}
\mathcal{A}_0^{(k+1)} &= \mathcal{A}_0^{(k)} - \mathcal{A}_{-1}^{(k)}\mathcal{K}^{(k)}\mathcal{A}_1^{(k)} - \mathcal{A}_1^{(k)}\mathcal{K}^{(k)}\mathcal{A}_{-1}^{(k)}, \quad \mathcal{K}^{(k)} = \left(\mathcal{A}_0^{(k)}\right)^{-1},\\
\mathcal{A}_{-1}^{(k+1)} &= -\mathcal{A}_{-1}^{(k)}\mathcal{K}^{(k)}\mathcal{A}_{-1}^{(k)},\\
\mathcal{A}_1^{(k+1)} &= -\mathcal{A}_1^{(k)}\mathcal{K}^{(k)}\mathcal{A}_1^{(k)},\\
\widehat{\mathcal{A}}_0^{(k+1)} &= \widehat{\mathcal{A}}_0^{(k)} - \mathcal{A}_1^{(k)}\mathcal{K}^{(k)}\mathcal{A}_{-1}^{(k)}.
\end{aligned} \tag{CR}$$

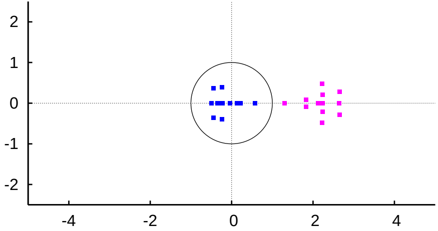
With some assumptions, (CR) converges to the solution of $\mathcal{A}_{-1} + \mathcal{A}_0 X + \mathcal{A}_1 X^2 = 0$ with smaller eigenvalues (in modulus)

1. spectral transformation $\mathcal{H} \mapsto I - t\mathcal{H}$ (shrink-and-shift)
2. apply (CR) to (UNI)

Example



Eigenvalues of $\mathcal{H}$



Eigenvalues of $I - t\mathcal{H}$

Fast cyclic reduction for (NT)

Cauchy-like structure

$$\begin{aligned}
\nabla_{\mathcal{D},\mathcal{D}}\mathcal{A}_{-1}^{(k)} =& \mathcal{A}_{-1}^{(k)} \begin{bmatrix} q \\ 0 \end{bmatrix} s_0^{(k)} + \mathcal{A}_0^{(k)} \begin{bmatrix} 0 \\ e \end{bmatrix} t_{-1}^{(k)} + u_0 \left[e^T, \ -q^T \right] \mathcal{A}_{-1}^{(k)}, \\
\nabla_{\mathcal{D},\mathcal{D}}\mathcal{A}_{0}^{(k)} =& \mathcal{A}_{-1}^{(k)} \begin{bmatrix} q \\ 0 \end{bmatrix} s_1^{(k)} + \mathcal{A}_0^{(k)} \begin{bmatrix} q \\ 0 \end{bmatrix} s_0^{(k)} + \mathcal{A}_0^{(k)} \begin{bmatrix} 0 \\ e \end{bmatrix} t_0^{(k)} \\
& + \mathcal{A}_1^{(k)} \begin{bmatrix} 0 \\ e \end{bmatrix} t_{-1}^{(k)} + u_0 \left[e^T, \ -q^T \right] \mathcal{A}_{0}^{(k)}, \\
\nabla_{\mathcal{D},\mathcal{D}}\mathcal{A}_{1}^{(k)} =& \mathcal{A}_{0}^{(k)} \begin{bmatrix} q \\ 0 \end{bmatrix} s_1^{(k)} + \mathcal{A}_1^{(k)} \begin{bmatrix} 0 \\ e \end{bmatrix} t_0^{(k)} + u_0 \left[e^T, \ -q^T \right] \mathcal{A}_{1}^{(k)},
\end{aligned}$$

Instead of updating $\mathcal{A}_{-1}^{(k)}$, $\mathcal{A}_{0}^{(k)}$, $\mathcal{A}_{1}^{(k)}$,

- update the above generators
- store and update the main diagonals of $\mathcal{A}_{-1}^{(k)}$, $\mathcal{A}_{1}^{(k)}$

Relations between SDA and CR

Theorem

SDA is CR applied to a different reduction (not only for (NT)!)

1. Spectral transformation
$$\mathcal{H} \mapsto \mathcal{H}_\gamma := (\mathcal{H} + \gamma I)^{-1}(\mathcal{H} - \gamma I) = \begin{bmatrix} I & -G_0 \\ 0 & F_0 \end{bmatrix}^{-1} \begin{bmatrix} E_0 & 0 \\ -H_0 & I \end{bmatrix}$$
2. Reduction to a quadratic eigenvalue problem
$$\begin{bmatrix} E_0 & 0 \\ -H_0 & I \end{bmatrix} u = \lambda \begin{bmatrix} I & -G_0 \\ 0 & F_0 \end{bmatrix} u$$
multiply the second block row by λ
$$\left(\begin{bmatrix} E_0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -I & G_0 \\ H_0 & -I \end{bmatrix} \lambda + \begin{bmatrix} 0 & 0 \\ 0 & F_0 \end{bmatrix} \lambda^2 \right) u = 0 \qquad \text{(SDA-U)}$$
3. (CR) on the unilateral equation associated to (SDA-U)

Much freedom, plenty of room for improvements

A new algorithm

Only when X is a square matrix — (NT) is ok

Idea: in the reduction step, try to make $\mathcal{H}$ triangular

1. shrink-and-shift $\mathcal{H} \to I - t\mathcal{H}$
2. conjugate to make the $(1,2)$ block nonsingular

$$\mathcal{H} \to \begin{bmatrix} I & M \\ 0 & I \end{bmatrix}^{-1} \mathcal{H} \begin{bmatrix} I & M \\ 0 & I \end{bmatrix} = \begin{bmatrix} * & -R(M) \\ * & * \end{bmatrix}$$

3. conjugate again to eliminate the $(1,1)$ block

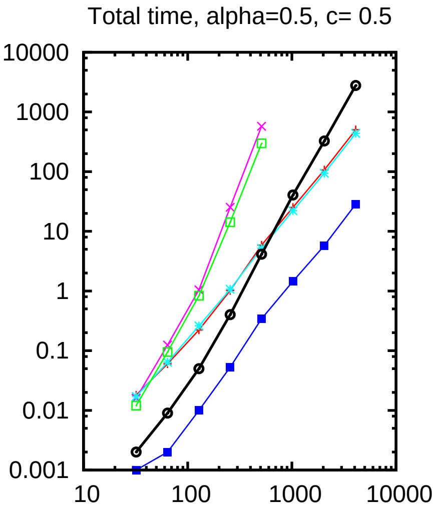
$$\mathcal{H} \to \begin{bmatrix} I & 0 \\ C^{-1}D & I \end{bmatrix}^{-1} \mathcal{H} \begin{bmatrix} I & 0 \\ C^{-1}D & I \end{bmatrix} = \begin{bmatrix} 0 & * \\ * & * \end{bmatrix}$$

4. A change of variables yields a $n \times n$ unilateral equation $\Rightarrow$ CR.

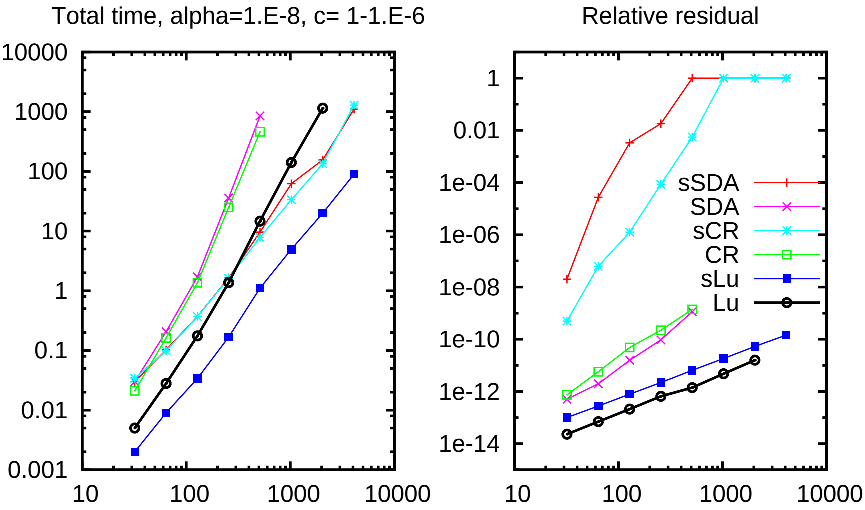
Cheaper to solve, $\frac{38}{3}n^3$/step instead of $\frac{64}{3}n^3$ (SDA)

Similar approach in [Bini–Iannazzo, '03]

Numerical results – noncritical case



Numerical results – quasi-critical case



Results and research lines

- **sLu** is the faster algorithm for (NT)
- **sSDA** and **sCR** could be useful for diagonal + rank r
- Better understanding of the algorithms, unified proofs
- Meaningful results not only for (NT), but for any NARE
- Ideas for new algorithms

Results and research lines

- **sLu** is the faster algorithm for (NT)
- **sSDA** and **sCR** could be useful for diagonal + rank r
- Better understanding of the algorithms, unified proofs
- Meaningful results not only for (NT), but for any NARE
- Ideas for new algorithms

Thanks for your attention!