SDA and cyclic reduction for a rank-structured algebraic Riccati equation

D. A. Bini[1] B. Meini[1] F. Poloni[1,2]

[1]Dipartimento di Matematica
Università di Pisa

[2]Scuola Normale Superiore, Pisa

Two-days of Numerical Linear Algebra
Bologna, 6–7 March, 2008

Outline

Motivations

Padua, Two-days of Numerical Linear Algebra 2007

F. Poloni: "Fast Newton method for an algebraic Riccati equation"

Research lines

- Fast SDA? The SDA iterates are generalized Cauchy-like as well

This year I am going to fill in the gap: fast $O(n^2)$ versions of two other algorithms for the same equation

Derivation and comparison between the algorithms

Algebraic Riccati equations

Nonsymmetric algebraic Riccati equation (NARE)

$$XCX - AX - XE + B = 0 \qquad \text{(NARE)}$$

$$X \in \mathbb{R}^{m \times n}, \text{ other matrices compatible}$$

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)

$$XCX - AX - XE + B = 0 \qquad \text{(NARE)}$$

$$X \in \mathbb{R}^{m\times n}, \text{ other matrices compatible}$$

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

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

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

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

Rank-structured NAREs

From a physics problem, we get

One-group neutron transport equation

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

D, Δ "positive" diagonal matrices, $e, q > 0$ vectors

(NT) is a NARE with rank structure:

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

Defined by $O(n)$ parameters; we can expect to find faster structured algorithms.

Solution algorithms

Brown: $O(n^3)$ per step, Green: $O(n^2)$ per step

Generic NARE

1. Newton's method [Guo–Laub, '99]
2. Cyclic Reduction [Ramaswami '99, Bini–Iannazzo–Latouche–Meini '05]
3. Structured doubling algorithm [Guo–Lin–Xu, '06]

Rank structured problem (NT)

4. Newton method on Lu's iteration [Lu '05]
5. Structured version of 1 and 4 [Bini–Iannazzo–P., preprint '06]
6. Secular equation [Mehrmann–Xu, preprint '07]
7. Structured version of 2 [Bini–Meini–P., preprint '08, this talk]
8. Structured version of 3 [Bini–Meini–P., preprint '08, this talk]

Cauchy-like matrices

Displacement operator

$$\nabla_{s,t}(M) := D_s M - M D_t$$

with $D_s = \mathrm{diag}(s)$, $D_t = \mathrm{diag}(t)$ diagonal matrices

M is said Cauchy-like if $\nabla_{s,t}(M)$ has low rank $r \iff$

$$M_{ij} = \frac{(U \cdot V)_{ij}}{s_i - t_j} \quad \text{whenever } s_i \neq t_j$$

U, V (*generators*) are $n \times r$, $r \times n$ matrices

We only keep in memory the generators, $2nr$ parameters

Usually one requires $s_i \neq t_j$ for all i,j

Instead, we will also need the case $s = t$ (*singular operator*):

nothing is known about the main diagonal of M

We keep in memory generators + diagonal (separately)

The GKO algorithm

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

Theorem (Gohberg–Kailath–Olshevsky)

During each step of Gaussian elimination $M \longrightarrow \begin{bmatrix} * & * \\ 0 & S \end{bmatrix}$, S *(the Schur complement) is Cauchy-like*

Instead of computing the elements of S $O(n^3)$, compute its generators $O(n^2)$

Singular operator case: hybrid strategy

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

Structured doubling algorithm (SDA)

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

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: $\mathcal{H}_\gamma = \mathcal{U}_0^{-1}\mathcal{L}_0$ with

$$\mathcal{U}_0 = \begin{bmatrix} I & -G_0 \\ 0 & F_0 \end{bmatrix}, \quad \mathcal{L}_0 = \begin{bmatrix} E_0 & 0 \\ -H_0 & I \end{bmatrix}$$

3. Implicit update $\mathcal{H}_\gamma^{2^k} = \mathcal{U}_k^{-1}\mathcal{L}_k$

The structured case

In the problem (NT), $\mathcal{H} = \mathcal{D} + uv$ (diagonal plus rank 1)
$\mathcal{H}_\gamma^{2^k}$ and $\mathcal{H}$ commute $\iff$

$$\mathcal{D}\mathcal{H}_\gamma^{2^k} - \mathcal{H}_\gamma^{2^k}\mathcal{D} = \mathcal{H}_\gamma^{2^k}uv - uv\mathcal{H}_\gamma^{2^k} \tag{1}$$

SDA preserves the Cauchy-like structure.

Need to compute explicit block generators? e.g. F_k:

pre- and post-multiply (1) by $\begin{bmatrix} 0 & F_k \end{bmatrix}$ and $\begin{bmatrix} 0 \\ F_k \end{bmatrix}$ to get

$$\Delta F_k - F_k \Delta = (H_k u_1 + u_2) v_2 F_k - F_k u_2 (v_1 + v_2 G_k)$$

Cauchy-like structure of SDA

In the same way,

$$
\begin{aligned}
DE_k - E_k D &= (u_1 + G_k u_2)v_1 E_k - E_k u_1(v_1 + v_2 H_k) \\
\Delta F_k - F_k \Delta &= (H_k u_1 + u_2)v_2 F_k - F_k u_2(v_1 + v_2 G_k) \\
DG_k + G_k \Delta &= (u_1 + G_k u_2)(v_1 + v_2 G_k) - E_k u_1 v_2 F_k \\
\Delta H_k + H_k D &= (H_k u_1 + u_2)(v_1 + v_2 H_k) - F_k u_2 v_1 E_k
\end{aligned}
\qquad \text{(GEN'S)}
$$

We can reconstruct the iterates from the eight vectors in blue/green (generators).
Instead of updating the matrices $O(n^3)$, update the generators $O(n^2)$
e.g.

$$F_{k+1} u_2 = F_k (I - H_k G_k)^{-1} F_k u_2$$

everything in the RHS can be computed using (GEN'S) and the generators at step k.
GKO for the inversion $O(n^2)$

Updating the diagonals

Problem: some of the operators are singular:

$$DE_k - E_k D = \dots$$
$$\Delta F_k - F_k \Delta = \dots \qquad \text{(GEN'S)}$$

We need to compute the diagonals of E_{k+1} and F_{k+1} as well.

Idea: after the generators update, we know:

- The off-diagonal elements of E_{k+1} and F_{k+1} (via the generators)
- $E_{k+1}u_1$ and $F_{k+1}u_2$ (two of the generators)

Easy to recover them:

$$(E_{k+1})_{jj} = \frac{(E_{k+1}u_1 - \text{off-diag}(E_{k+1})u_1)_j}{(u_1)_j}$$

Issue: stability?

Outline of cyclic reduction (CR)

1. Spectral transformation (as in SDA)
2. Transform (NARE) to the unilateral equation

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

3. Solve (UNI) via cyclic reduction.

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

$$\begin{aligned} S_{k+1} &= S_k - R_k S_k^{-1} T_k - T_k S_k^{-1} R_k \\ R_{k+1} &= -R_k S_k^{-1} R_k \\ T_{k+1} &= -T_k S_k^{-1} T_k \\ \widehat{S}_{k+1} &= \widehat{S}_k - T_k S_k^{-1} R_k, \quad \widehat{S}_0 = S_0 \end{aligned} \tag{CR}$$

Converges quadratically to the spectral minimal solution of $R_0 + S_0 Y + T_0 Y^2 = 0$

Interpretation of CR [Bini–Latouche–Meini '05]:

- Let $\varphi^{(k)}(z) = R_k z^{-1} + S_k + T_k z$
- Let $\psi^{(k)}(z) = \varphi^{(k)}(z)^{-1}$
- (CR) can be seen as the update $\psi^{(k+1)} = \text{even}\left(\psi^{(k)}\right)$

$$\text{even}(\psi) = \cdots + \psi_{-4} z^{-2} + \psi_{-2} z^{-1} + \psi_0 + \psi_2 z + \psi_4 z^2 + \cdots$$

The structured case

For the low-rank problem (NT),

$$\varphi^{(0)} = D(z) + uv(z)$$

is diagonal plus rank 1

...some computations lead to...

$$\nabla_{\mathcal{D},\mathcal{D}}\psi^{(0)} = u_1 v_1(z) + u_2 v_2(z) + u_3(z) v_3$$

This structure is preserved under even$(\cdot)$ $\Rightarrow$ $\nabla_{\mathcal{D},\mathcal{D}}\psi^{(k)}$ has rank 3 for all k

...even more computations lead to...

Cauchy-like structure of CR

Cauchy-like structure

$$\nabla_{\mathcal{D},\mathcal{D}} R_k = R_k u_1 s_0^{(k)} + S_k u_2 t_{-1}^{(k)} + u_0 v_3 R_k,$$
$$\nabla_{\mathcal{D},\mathcal{D}} S_k = R_k u_1 s_1^{(k)} + S_k u_1 s_0^{(k)} + S_k u_2 t_0^{(k)} + T_k u_2 t_{-1}^{(k)} + u_0 v_3 S_k,$$
$$\nabla_{\mathcal{D},\mathcal{D}} T_k = S_k u_1 s_1^{(k)} + T_k u_2 t_0^{(k)} + u_0 v_3 T_k,$$

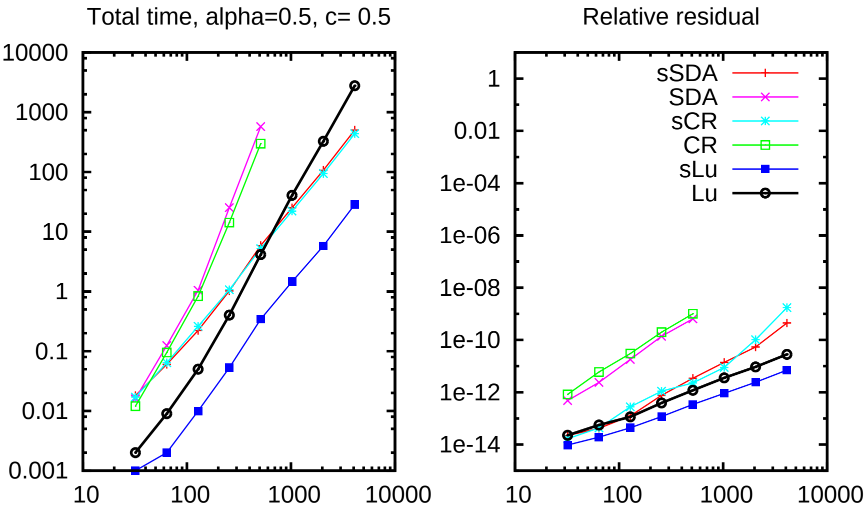
R_k, S_k, T_k have size $n+m$, but there are some zero or known blocks we can skip

We can reconstruct the iterates from

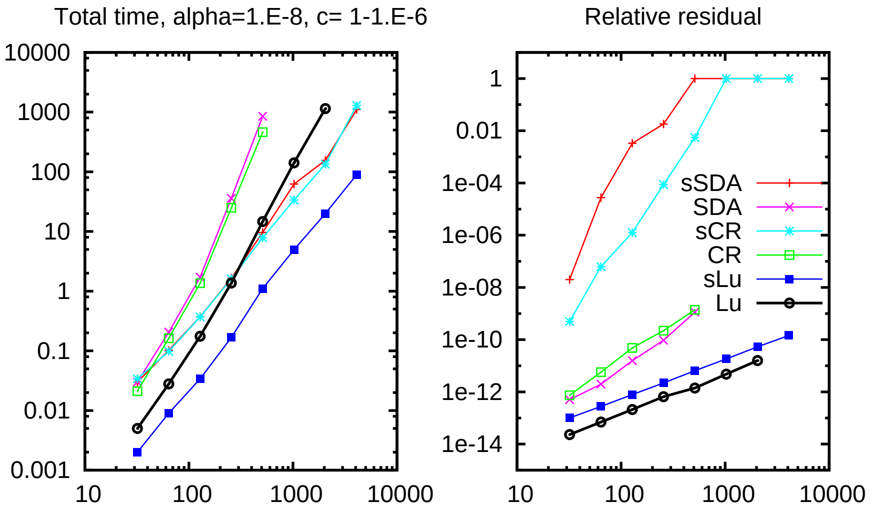
- 8 vectors of length n or m
- 2 diagonals

Proceed as in SDA: update vectors and diagonals

Numerical results – noncritical case



Numerical results – quasi-critical case



To sum up...

- Structural analysis (for Cauchy-like input) of SDA and CR
  - Better understanding of the algorithms
- Developed structured versions of SDA and CR
- Faster than nonstructured algorithms
- Not as fast as structured Lu/Newton
- Loss of precision in near-to-critical cases
  - Stabler ways to recover diagonal of iterates?
- Can be generalized to diag+rank r; scales as $O(n^2 r)$
  - Lu/Newton would scale as $O(n^2 r^2)$
  - Needed in applications? Solution “looking for a problem”

Another kind of fast SDA

(Thanks to Antonio for the joke)