# Large-scale methods for Lyapunov equations

We give a hint of the methods used for large-scale equations.

We focus on Lyapunov equations, $AX + XA^* + BB^* = 0$.
(Then we can solve CAREs using Newton's method.)

Assumptions: $A$ large and sparse with $\Lambda(A) \subset LHP$. $B \in \mathbb{R}^{n\times m}$, with $m \ll n$.

Actually, we may suppose $B = b \in \mathbb{R}^n$ without loss of generality: a rank-$m$ matrix is the sum of $m$ rank-1 matrices, and the equation is linear.

Assume $A$ symmetric, normal or 'almost normal'. The algorithms often work for generic $A$, but the analysis works better for normal matrices.

Roadblock: the solution $X$ is dense!
Solution: often, $X \approx ZZ^*$ with a tall thin $Z$: it has decaying singular values and low numerical rank.

# ADI (alternating-direction implicit iteration)

Let $\tau > 0$, so that $\Lambda(A - \tau I) \subset LHP$. One can rewrite $AX + XA^* + bb^* = 0$ as

$$(A - \tau I)X(A - \tau I)^* - (A + \tau I)X(A + \tau I)^* - 2\tau bb^* = 0$$

or (with $c(x) = \frac{x+\tau}{x-\tau}$)

$$X - c(A)Xc(A)^* = 2\tau(A - \tau I)^{-1}bb^*(A - \tau I)^{-*}.$$

This suggests the fixed-point iteration

$$X_0 = 0, \quad X_k = c(A)X_{k-1}c(A)^* + 2\tau(A - \tau I)^{-1}bb^*(A - \tau I)^{-*}.$$

In addition, we can change the value of $\tau$ at each iteration: $c_k(x) = \frac{x+\tau_k}{x-\tau_k}$,

$$X_k = c_k(A)X_{k-1}c_k(A)^* + 2\tau_k(A - \tau_k I)^{-1}bb^*(A - \tau_k I)^{-*}.$$

# Low-rank ADI

$$X_0 = 0, \quad X_k = c_k(A) X_{k-1} c_k(A)^* + 2\tau_k (A - \tau_k I)^{-1} b b^* (A - \tau_k I)^{-*}.$$

can be rewritten in terms of the 'low-rank factor' of $X_k = Z_k Z_k^*$:

$$Z_0 = 0, \quad Z_k = \left[ c_k(A) Z_{k-1} \quad \sqrt{2\tau_k}(A - \tau_k I)^{-1} b \right].$$

The cumulative effect of $k$ steps is (with $d_k(x) = \sqrt{2\tau_k}(x - \tau_k)^{-1}$)

$$Z_k = \left[ c_k c_{k-1} \dots c_2 d_1(A) b \quad \cdots \quad c_k d_{k-1}(A) b \quad d_k(A) b \right].$$

# Low-rank ADI

$$Z_k = \begin{bmatrix} c_k c_{k-1} \dots c_2 d_1(A)b & \cdots & c_k d_{k-1}(A)b & d_k(A)b \end{bmatrix},$$

We can compute the same quantity “starting from the right”:

$$\begin{aligned} Z_k &= \begin{bmatrix} d_1 c_2 \dots c_{k-1} c_k(A)b & \cdots & d_{k-1} c_k(A)b & d_k(A)b \end{bmatrix} \\ &= \begin{bmatrix} v_1 & v_2 & \cdots & v_k \end{bmatrix}. \end{aligned}$$

Since $c_j(x) = 1 + \sqrt{2\tau_j} d_j(x)$, we can compute this iteratively (details omitted).

### Low-rank ADI

$$v_1 = \sqrt{2\tau_1}(A - \tau_1)^{-1} b, \quad v_j = \frac{\sqrt{2\tau_j}}{\sqrt{2\tau_{j-1}}}(v_j + (\tau_{j-1} + \tau_j)(A - \tau_j I)^{-1} v_j).$$

# ADI: convergence

ADI residual:

$$X_k - X_* = c_k(A)(X_{k-1} - X_*)c_k(A)^* = \cdots = g(A)(X_0 - X_*)g(A)^*,$$

where $g(x) = \prod_{i=j}^{k} \frac{x-\tau_j}{x+\tau_j}$.

Convergence speed depends on the choices of $\tau_j$. Intuitively: good if $A + \tau_j I$ is small and $A - \tau_j I$ is large. This suggests taking $\tau_j$ as (some of) the eigenvalues of $A$.

If $A = V\Lambda V^{-1}$, then

$$\|g(A)\| \leq \kappa(V) \max_{\lambda \in \Lambda(A)} \prod_{j=0}^{k} \frac{|\lambda - \tau_j|}{|\lambda + \tau_j|}.$$

How to choose $\tau_j$'s that make this small? Easy if $A$ has few / clustered eigenvalues.

# ADI convergence

$$\eta_k = \min_{\tau_0,\dots,\tau_k} \max_{\lambda \in \Lambda(A)} \prod_{j=0}^{k-1} \frac{|\lambda - \tau_j|}{|\lambda + \tau_j|}.$$

In general, tricky approximation theory problem. Typical approach: find an enclosing region for the eigenvalues of $A$ (for instance, if $A = A^*$, all eigenvalues are in $[\lambda_{\min}, \lambda_{\max}]$).

Then, we look for a polynomial that is 'small' on $[\lambda_{\min}, \lambda_{\max}]$ and 'large' on $[-\lambda_{\max}, -\lambda_{\min}]$.

In many cases, $\eta_k \sim r^k$ for a certain $r < 1$.

In particular, the solution $X_*$ is "well-approximated" by the matrix $X_k$ of rank $k$.
Consequence The singular values of $X$ decay as $\sim r^k$, low numerical rank.

(Good thing, because otherwise the problem would be hopeless.)

# Residual computation

For $X_k = Z_k Z_k^*$, with $Z_k \in \mathbb{R}^{n\times k}$, we have

$$AZ_kZ_k^* + Z_kZ_k^*A^* + BB^* = \begin{bmatrix} Z_k & AZ_k & B \end{bmatrix} \begin{bmatrix} 0 & I & 0 \\ I & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} Z_k & AZ_k & B \end{bmatrix}^* .$$

Using QR or SVD of the tall thin $\begin{bmatrix} Z_k & AZ_k & B \end{bmatrix}$, we can compute residual norms in $O(nk^2)$.

# Rational Arnoldi

The computed $Z_k$ has columns of the form $r(A)b$, where $r(x) = q(x)/p(x)$, denominator $p(x) = (x - \tau_1)(x - \tau_2)\dots(x - \tau_k)$.

Hence, our approximation $Z_k$ lives in a rational Arnoldi subspace

$$K_q(A, b) = \{p(A)^{-1}q(A)b\colon \deg q < k\} = p(A)^{-1}K_k(A, b).$$

Idea: first compute this subspace, then solve the projected equation.

# Galerkin Projection

Given an orthonormal basis $U_k$ of $K_q(A, b)$:

1. Set $X_k = U_k Y_k U_k^*$;
2. Assume 'orthogonal residual': $U_k^*(AX_k + X_k A^* + BB^*)U_k = 0$.

Produces a projected Lyapunov equation

$$(U_k^* A U_k)Y + Y(U_k^* A U_k)^* + U_k^* BB^* U_k = 0.$$

Difficulty 1 Even if $A$ stable, $U_k^* A U_k$ is not necessarily so (unless $A$ symmetric!).
Difficulty 2 (main one, shared with ADI): good pole selection.