Methods for large-scale control systems

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

What does a large-scale control system look like?
Example: the heat equation: finite-difference discretization of a 2D or 3D structure, possibly with a nontrivial shape.

- Large, sparse $A \in \mathbb{R}^{n \times n}$ produced by the discretization (state evolves 'locally').
- $B \in \mathbb{R}^{n \times m}$ with $m \ll n$ (the control acts only on a few points). Hence $G = BR^{-1}B^*$ has low rank.
- $Q = C^*C$ is also often taken to be low-rank (energy based on 'output' values measured in a few points).

Large-scale Lyapunov 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 $m \ll n$.

In addition, 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 and full-rank ($X \succeq 0$)!
Solution: often, $X \approx ZZ^*$ with a tall thin Z: it has decaying singular values and low numerical rank (we will see why).

Time discretization

Underlying idea: let's switch from a continuous-time control system to a discrete-time one; in particular, we would like to use the midpoint method:

$$\dot{x} = Ax + Bu$$

is discretized to

$$\frac{x_{k+1} - x_k}{h} = \frac{1}{2}\left(Ax_k + Bu_k + Ax_{k+1} + Bu_{k+1}\right),$$

i.e.,

$$x_{k+1} = (I - \tfrac{h}{2}A)^{-1}(I + \tfrac{h}{2}A)x_k + (I - \tfrac{h}{2}A)^{-1}B(u_k + u_{k+1})\tfrac{h}{2}.$$

This specific method is particularly nice, because it preserves stability: the open-loop system $\dot{x} = Ax$ is stable iff $x_{k+1} = (I - \frac{h}{2}A)^{-1}(I + \frac{h}{2}A)x_k$ is so.

Lemma For each $\tau > 0$, the map $c(x) = \frac{x+\tau}{x-\tau}$ is such that $c(LHP) =$ unit disc.

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

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

a Stein equation. We can solve it with the fixed-point iteration

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

Recall that $c(LHP) =$ unit disc; hence $\Lambda(c(A)) \subset$ unit disc and the iteration converges.

Low-rank ADI

Setting $\hat{A} = c(A)$ and $\hat{b} = \sqrt{2\tau}(A - \tau I)^{-1} b$, we have

$$X_k = \hat{b}\hat{b}^* + \hat{A}\hat{b}\hat{b}^*\hat{A} * + \hat{A}^2\hat{b}\hat{b}^*\hat{A}^{2*} + \cdots + \hat{A}^k\hat{b}\hat{b}^*\hat{A}^{k*}.$$

Or, in terms of its low-rank factor

$$Z_k = \begin{bmatrix} \hat{b} & \hat{A}\hat{b} & \hat{A}^2\hat{b} & \dots & \hat{A}^k\hat{b} \end{bmatrix}, \quad X_k = Z_k Z_k^*.$$

We can get faster convergence by changing the value of τ at each step: with

$$c_k(x) = \frac{x + \tau_k}{x - \tau_k}, \quad d_k(x) = \sqrt{2\tau_k}\frac{1}{x - \tau_k},$$

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

Low-rank ADI

After k steps we have

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

We can compute the same quantity "starting from the right":

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

Since $c_j(x) = \frac{1}{\sqrt{2\tau_j}}(x+\tau_j)d_j(x)$, we can compute the v_j's iteratively.

Low-rank ADI

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

One can also use complex shifts (details omitted, $\overline{\tau}_j$s appears).

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 τ_j. Intuitively: good if $A + \tau_j I$ is small and $A - \tau_j I$ is large. This suggests taking τ_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 τ_j's that make this norm small? Easy if A has few / clustered eigenvalues.

ADI convergence

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

The optimum depends on $\Lambda(A)$.
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, look for a polynomial that is 'small' on $[\lambda_{\min}, \lambda_{\max}]$ and 'large' on $[-\lambda_{\max}, -\lambda_{\min}]$.

Deep, classical problem from approximation theory; explicit solutions can be constructed from elliptic functions. It is known that in the optimum $\eta_k \sim r^k$ for a certain $r < 1$ that depends on $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$.

Consequence Since $\|X_* - X_k\| \sim r^k$, and $\operatorname{rk} X_k = k$, it follows that $\sigma_{k+1}(X) \lesssim r^k$, so X has low numerical rank.

Residual computation

Detail As a natural stopping criterion, we may use the residual $AZ_kZ_k^* + Z_kZ_k^*A^* + BB^*$, but how to compute it without assembling large matrices?
For $X_k = Z_kZ_k^*$, 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 a QR of the tall thin $\begin{bmatrix} Z_k & AZ_k & B \end{bmatrix}$, we can compute this norm in $O(nk^2)$.

Rational Arnoldi

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

Hence, our approximation Z_k lives in a rational Arnoldi subspace

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

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 $\Lambda(A) \subset LHP$, the same property does not always hold for $U_k^* A U_k$.
Recall: the eigenvalues of $A_k = U_k^* A U_k$ are in the field of values of A, which is $\operatorname{hull} \Lambda(A)$ for normal A, but larger (possibly by much) for non-normal A.
Difficulty 2 (main one, shared with ADI): good pole selection.