Note Title 2021-05-13

Problemi di controllo di larga scala

$\dot{x} = Ax + Bu$ $\quad A \in \mathbb{R}^{n\times n}$ larga e sparsa (o comunque con prodotto matvec $v \mapsto Av$ veloce e soluzione di sistemi lineari $b \mapsto A^{-1}b$, $b \mapsto (A-\tau I)^{-1}b$)

$B \in \mathbb{R}^{n\times m}$

$Q = C^TC \quad C \in \mathbb{R}^{p\times n} \quad m, p \ll n \qquad G = BR^{-1}B^T$

Vogliamo risolvere problemi di controllo ottimale $A^TX + XA + Q - XGX = 0$

Usiamo Newton → ci basta saper risolvere eq. di Lyapunov

$$\underbrace{(A - GX_k)}_{\text{sparsa+rk basso}}{}^T X_{k+1} + X_{k+1}(A - GX_k) + \underbrace{Q + XGX_k}_{\text{rk basso}} = 0 \qquad (X_k \text{ noto})$$

Semplificazione: mi basta saper risolvere eq. di Lyapunov

$AX + XA^T + BB^T = 0$

Semplificazione: mi basta trattare il caso B è un vettore, $B = b$

$$\boxed{AX + XA^T + bb^T = 0} \qquad (L) \qquad \Leftrightarrow (I\otimes A^T + A^T \otimes I)\,\mathrm{vec}\,X = \mathrm{vec}(bb^T)$$

(L'eq. di Lyapunov è lineare, quindi per risolvere $AX + XA^T + b_1b_1^T + b_2b_2^T + \dots + b_mb_m^T = 0$ mi basta sommare le soluzioni X_k di $AX_k + X_kB^T = b_kb_k^T$)

Lemma: X risolve (L) $\Leftrightarrow$ X risolve

$$X - \hat{A}X\hat{A}^T - \hat{b}\hat{b}^T = 0 \qquad (S)$$

con $\hat{A} = \underline{(A-\tau I)^{-1}(A+\tau I)}$ $\quad \hat{b} = \sqrt{2\tau}\,\underline{(A-\tau I)^{-1}b}$ $\qquad \tau > 0$

(si estende anche a $\tau \in RHP$)

($\hat{A}, \hat{b}$ del sistema dinamico $x_{k+1} = \hat{A}x_k + \hat{B}u_k$ ottenuto con discretizzazione con il met. dei trapezi $\dot{x} = Ax + Bu$)

dim.

(S) $\quad X - (A-\tau I)^{-1}(A+\tau I)\,X\,(A+\tau I)^T(A-\tau I)^{-T} - 2\tau\,(A-\tau I)^{-1}bb^T(A-\tau I)^{-T}$

$\iff$

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

$\iff$

$$\cancel{AXA^T} - \tau AX - \tau XA^T + \cancel{\tau^2 I} - \cancel{AXA^T} - \tau AX - \tau XA^T - \cancel{\tau^2 I} - 2\tau bb^T = 0$$

$$-2\tau\big(\underbrace{AX + XA^T + bb^T}_{(L)}\big) = 0$$

Posso risolvere la (S) con un metodo di punto fisso

$$(S):\ X = \hat b\hat b^T + \hat A X \hat A^T$$

$$X_0 = 0 \qquad X_{k+1} = \hat b\hat b^T + \hat A X_k \hat A^T = \hat b\hat b^T + \hat A\left(\hat b\hat b^T + \hat A X_{k-1}\hat A^T\right)\hat A^T = \dots$$

$$\dots = \hat b\hat b^T + \hat A\hat b\hat b^T\hat A^T + \hat A^2\hat b\hat b^T\hat A^{2T} + \dots + \hat A^{k+1}\hat b\hat b^T\hat A^{(k+1)T}$$

Cioè, $X_k = Z_k Z_k^T$

$$Z_k = \left[\hat b \quad \hat A b \quad \hat A^2 b \quad \cdots \quad \hat A^{k+1}\hat b\right]$$

se $\Lambda(A) \subset LHP$, allora $\Lambda(\hat A) \subset$ disco unitario, e $\lim_{k\to\infty} \hat A^k = 0$

ADI iteration (alternating-direction implicit iteration)

Per farlo convergere più velocemente è opportuno cambiare τ a ogni iterazione: all'iterazione k usate la relazione

$$c_k(x) = \frac{x+\tau_k}{x-\tau_k}$$

$$X_k = c_k(A)X_{k-1}c_k(A) + d_k(A)bb^Td_k(A)^T \qquad d_k(x) = \sqrt{2\tau_k}\,\frac{1}{x-\tau_k}$$

$$Z_k = \Big[\underbrace{d_k(A)b}_{v_1} \quad \underbrace{c_k d_{k-1}(A)b}_{v_2} \quad \underbrace{c_kc_{k-1}d_{k-2}(A)b}_{v_3} \cdots \underbrace{c_kc_{k-1}\cdots c_2 d_1(A)b}_{v_{k-1}}\Big]$$

$$v_1 = \sqrt{2\tau_k}(A-\tau_k I)^{-1}b \qquad v_2 = (A-\tau_k I)^{-1}(A+\tau_k I)\sqrt{2\tau_{k-1}}(A-\tau_{k-1}I)^{-1}b$$

$$= (A-\tau_{k-1}I)^{-1}(A+\tau_k I)\frac{\sqrt{2\tau_{k-1}}}{\sqrt{2\tau_k}}\underbrace{\sqrt{2\tau_k}(A-\tau_k I)^{-1}b}_{v_1}$$

$$v_{k+1-j} = \frac{\sqrt{2\tau_{j-1}}}{\sqrt{2\tau_j}}(A-\tau_{j-1}I)^{-1}(A+\tau_j I)\,v_{k+2-j}$$

Se X_* è la soluzione di (S) e (L)

$$\underbrace{X_k - X_*}_{E_k} = C_k(A)X_{k-1}C_k(A)^T + \cancel{d_k(A)bb^Td_k(A)^T} - C_k(A)X_*C_k(A)^T - \cancel{d_k(A)bb^Td_k(A)^T}$$

$$E_k = C_k(A)E_{k-1}C_k(A)^T = C_k(A)C_{k-1}(A)E_{k-2}C_{k-1}(A)^TC_k(A)^T = \dots$$

$$\dots g(A)E_0\,g(A)^T$$

$$g(x) = C_k(x)C_{k-1}(x)\cdots C_1(x) = \frac{(x+\tau_k I)(x+\tau_{k-1}I)\cdots(x+\tau_1 I)}{(x-\tau_k I)(x-\tau_{k-1}I)\cdots(x-\tau_1 I)}$$

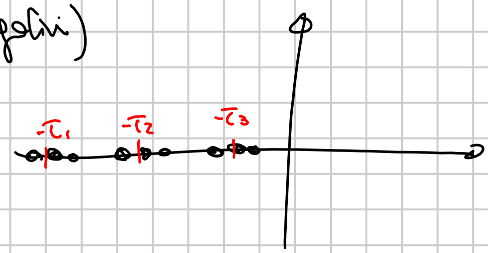
più $\|g(A)\|$ è piccola, più si ha convergenza velocemente

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

$$\|g(A)\| = \|V\,g(\Lambda)V^{-1}\| \le \kappa(V)\,\underline{\max_{\lambda\in\Lambda(A)} \prod_{j=1}^{k} \frac{|\lambda+\tau_j|}{|\lambda-\tau_j|}}$$

Se A avesse solo k autovalori distinti, posso scegliere $\tau_j = -\lambda_j$
in modo che $g(\lambda)=0$ per ogni autovalore λ
(almeno se A ha solo autoval. reali negativi)

Se A ha k "cluster" di autovalori,
riesco ad avere numeratore "piccolo"



Scelta ottimale:

$$\eta_k = \min_{\tau_1,\dots\tau_k}\ \max_{\lambda\in\Lambda(A)} \prod \frac{|\lambda+\tau_j|}{|\lambda-\tau_j|}$$

(Tipicamente impossibile da calcolare, perché serve sapere $\Lambda(A)$)

Euristica: mi approssimo un po' di autovalori di A (tipicamente i più grandi e i più piccoli) con qualche passo di Arnoldi su A, A^{-1} e scelgo i τ_j tra gli autoval. calcolati in modo da minimizzare questa funzione razionale; genero una sequenza finita $[\tau_1, \tau_2, \dots \tau_m]$, e poi li ciclo, $\tau_{m+1} = \tau_1$, $\tau_{m+2} = \tau_2$, ...

Nel caso di τ fisso, $\qquad \hat{A} = (A-\tau I)^{-1}(A+\tau I)$

$$Z_k = [\hat{b} \quad \hat{A}\hat{b} \quad \dots \quad \hat{A}^k \hat{b}]$$ è una base di $K(\hat{A}, \hat{b})$

Nel caso τ_k variabili,

$$Z_k = [d_k(A)b \quad \dots \quad c_k c_{k-1} \cdots d_1(A) b]$$ è una base di $K_{q,k}(A,b)$

spazio di Krylov razionale con polinomio $q(x) = (x-\tau_1)(x-\tau_2)\cdots(x-\tau_k)$

$K_{q,k}(A,b) = \{ r(A)b : r(x) = \frac{p(x)}{q(x)}$ con q come denom. e $\deg p < k \}$

Idea: prima calcolo una base $Q_k \in \mathbb{R}^{n\times k}$ di un opportuno spazio di Krylov,

poi pongo $X_k = Q_k Y_k Q_k^T$

$$Q_k^T(AX_k + X_k A^T + bb^T)Q_k = 0 \qquad (*)$$ "Galerkin projected equation"

$0 = \underbrace{Q_k^T A Q_k}_{A_k} Y_k \cancel{Q_k^T Q_k} + \cancel{Q_k^T Q_k} Y_k \underbrace{Q_k^T A^T Q_k}_{A_k^T} + Q_k^T b b^T Q_k$ è un'eq. di Lyapunov "proiettata" per Y di dimensione $k \times k$

$$A_k Y_k + Y_k A_k^T + d_k d_k^T = 0$$

Algoritmo:

1) Costruisco Q_k base di uno sp. di Krylov
2) Risolvo $(*)$ per trovare Y_k
3) $X_k = Q_k Y_k Q_k^*$ è un'approssimazione della soluzione

Punti critici... 1) trovare poli buoni τ_k

2) Se A non simmetrica, può succedere che $Q_k^T A Q_k = A_k$ abbia $\Lambda(A_k) \not\subset LHP$, e in questo caso l'unicità della sol. non è più garantita

Nota:

Errore nell'esempio numerico con ADI visto a lezione:

Matlab interpreta `>> bhat = sqrt(2*tau) * (A - tau*eye(n)) \ b`

come `>> bhat = (sqrt(2*tau) * (A - tau*eye(n))) \ b`

quindi divide anziché moltiplicare per $\sqrt{2\tau}$

Una volta sistemato questo funziona tutto.