Soluzione numerica di equazioni di Riccati

Note Title 2024-05-29

(R) $\quad A^* X + XA + Q - XGX = 0 \qquad A, X, G, Q \in \mathbb{C}^{n\times n}$

$Q, G \succeq 0$

$\exists!$ sol. X t.c. $\Lambda(A-GX) \subset LHP \qquad (A,G), (A^*,Q)$ stabilizz.

$X = X^* \succeq 0$

$$X \text{ soluzione di } (R) \iff \begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix} (A - GX)$$

Risolvere (R) $\iff$ trovare un sott. invariante di $H = \begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix}$

più precisamente, il sottosp. invariante associato al LHP

1) Metodo di Newton su $F(X) = A^* X + XA + Q - XGX$

$\mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$

Ci serve la derivata di Fréchet di F:

$$F(X+H) = A^*(X+H) + (X+H)A + Q - (X+H)G(X+H)$$

$$= F(X) + \underbrace{A^* H + HA - HGX - XGH}_{} + O(H^2)$$

$$L_{F,X}[H] = (A-GX)^* H + H(A-GX)$$

Metodo di Newton multivariato:

1) risolviamo $(A - GX_k)^* H + H(A - GX_k) = F(X_k)$ per trovare H,

2) $X_{k+1} = X_k - H$

<u>Oss:</u> • Ad ogni passo, devo risolvere un'eq. di Lyapunov.

- La derivata di Fréchet nella sol. esatta L_{F,X_*}

si scrive in forma di Kronecker come

$$\hat{K} = (A - GX_*)^T \otimes I + I \otimes (A - GX_*)^*$$

Autovalori $\lambda_i + \bar{\lambda}_j$, dove $\{\lambda_1, \dots \lambda_n\} = \Lambda(A - GX_*)$

$\lambda_i + \bar{\lambda}_j \in LHP \Rightarrow$ la Jacobiana è invertibile $\Rightarrow$ Newton converge quadraticamente.

- Problema non banale: trovare un punto iniziale che converge alla sol. stabilizzante X_*
- Usato spesso solo come "refinement" dopo un altro algoritmo

---

2) Metodo di Schur

Forma di Schur $\to$ sottosp. invariante

$U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$ base di un sottosp. invariante $\Rightarrow X = U_2 U_1^{-1}$

$\mathcal{U} = \text{span}\left(\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}\right) = \text{span}\left(\begin{bmatrix} I \\ X \end{bmatrix}\right)$

$U = \begin{bmatrix} U_1 & U_3 \\ U_2 & U_4 \end{bmatrix}$ dato da Schur

$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$ base di $\mathcal{U}$ $\quad X = U_2 U_1^{-1}$

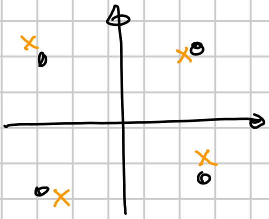
Stab. all'indietro di Schur $\Rightarrow$ $\hat{U}_1, \hat{U}_2$ sono le U_1, U_2 corrette

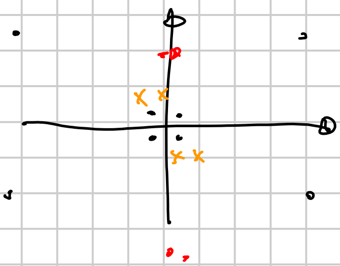
per una piccola perturbazione $\hat{H}$ $\quad \frac{\|\hat{H} - H\|}{\|H\|} = O(u)$

⚠ Non è detto (e in generale è falso) che

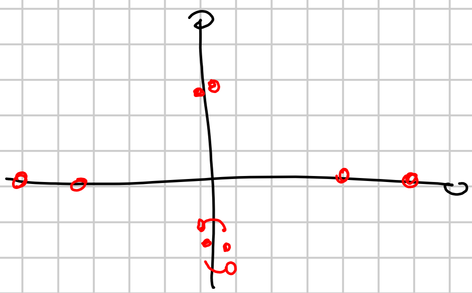
$\hat{H} = \begin{bmatrix} \hat{A} & -\hat{G} \\ -\hat{Q} & -\hat{A}^* \end{bmatrix}$: il blocco (2,2) non sarà meno il trasp. coniug. del blocco (1,1), e i blocchi $(1,2)$ e $(2,1)$ non saranno per forza $\neq 0$.

$\Rightarrow$ gli autovalori non vengono sempre in coppie esattamente simmetriche





In un problema con $\frac{Re(\lambda_i)}{\|H\|} \sim O(u)$, queste perturbazioni possono spostare autovalori dal LHP al RHP.



È possibile fare un metodo di Schur che preservi la struttura Hamiltoniana? Sì, ma con attenzione: bisogna usare trasformazioni ortogonali anche rispetto al prod. scalare $\langle u,v\rangle = u^* J v$, e lavorare implicitamente su H^2 senza formarla.

---

3) Iterazione di Newton per il segno.

Idea: calcolo $\mathrm{sign}(H)$, e questo mi permette di rivelare il sott. invariante stabile $U = \mathrm{Ker}(\mathrm{sign}(H) + I)$

Difetti: se

$$H = V \begin{bmatrix} J_- & \\ & J_+ \end{bmatrix} V^{-1}$$ forma di Jordan con $\Lambda(J_-) \subset LHP$, $\Lambda(J_+) \subset RHP$

allora

$$\text{sign}(H) = V \begin{bmatrix} -I & \\ & I \end{bmatrix} V^{-1} \qquad \text{sign}(H)+I = V \begin{bmatrix} 0 & \\ & 2I \end{bmatrix} V^{-1}$$

$$\text{Ker}(\text{sign}(H)+I) = \text{span}(\underbrace{v_1, v_2, \dots v_n}_{\text{prime } n \text{ colonne di } V}) = \text{sott. inv. stabile di } H.$$

$= \text{Im}(\text{sign}(H) - I)$, con un conto analogo.

Per calcolare $\text{sign}(H)$, usiamo l'iterazione di Newton per il segno:

$$H_0 = H \qquad H_{k+1} = \frac{1}{2}\left(H_k + H_k^{-1}\right)$$

- posso incorporare scaling: $H_0 = \alpha H$ con $\alpha > 0$ scelto in modo da "bilanciare" autovalori grandi e piccoli

$$\frac{1}{2}\left(x + \frac{1}{x}\right)$$

Osservazione importante: ad ogni passo, il metodo produce iterate che sono di nuovo Hamiltoniane.
Questo segue da:

Lemma:

1) H Hamiltoniana $\Rightarrow$ H^{-1} Hamiltoniana
2) H_1, H_2 Hamiltoniane $\Rightarrow$ $\alpha H_1 + \beta H_2$ è Hamiltoniana (α, β reali)

Dim: Ricordiamo che H Hamilt. $\Leftrightarrow \begin{bmatrix} A & -G \\ -H & -A^* \end{bmatrix}$

con G, H simmetriche $\iff$ $-H^* J = JH$ $\qquad J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$

$$-\langle Hu, v \rangle = -u^* H^* J v = u J H v = \langle u, Hv \rangle$$

1) $-H^* J = JH$ moltiplico per H^{-*}, H^{-1} e ottengo

$$-JH^{-1} = H^{-*} J \iff H^{-1} \text{ è Hamiltoniana}$$

2) $-H_1^* J = JH_1$ $\quad -H_2^* J = JH_2$ $\Rightarrow$ $-(\alpha H_1 + \beta H_2)^* J = J(\alpha H_1 + \beta H_2)$.

Quindi: H_0 Hamiltoniana $\Rightarrow$ $H_1 = \frac{1}{2}(H_0 + H_0^{-1})$ Hamiltoniana

Struttura preservata $\Rightarrow$ anche se H_0 è molto mal condizionata, il metodo produce $H_* = \lim_{k \to \infty} H_k$ che è Hamiltoniana

$\Rightarrow$ la X prodotta è sempre simmetrica, e H_* ha sempre n autoval. 1 e n autoval. -1.

⚠ In aritmetica di macchina, la H_1 prodotta potrebbe non essere esattamente Hamiltoniana!

"inv" non preserva esattamente la struttura Hamiltoniana, però preserva esattamente la simmetria.

<u>Idea</u>: cambio l'iterazione in modo che lavori con matrici simmetriche

<u>Lemma</u>: H Hamiltoniana $\iff$ JH Hermitiana

$$-H^* J = JH \qquad\qquad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$(JH)^* = H^* J^* = H^*(-J) = JH$$

$$JH = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix} = \begin{bmatrix} -Q & -A^* \\ -A & G \end{bmatrix} \qquad \square$$

Definiamo $Z_k := J H_k$ e riscriviamo l'iterazione in funzione degli Z_k:

$$H_{k+1} = \frac{1}{2}\left(H_k + H_k^{-1}\right)$$

$$J H_{k+1} = \frac{1}{2}\left(J H_k + J H_k^{-1}\right)$$

$$Z_k^{-1} = H_k^{-1} J^{-1}$$

$$Z_k^{-1} J = H_k^{-1}$$

$$\boxed{Z_{k+1} = \frac{1}{2}\left(Z_k + J Z_k^{-1} J\right)}$$

In Matlab, inv(Z_k) produce sempre una matrice esattamente simmetrica, e anche le altre operazioni restituiscono sempre matrici simmetriche $\Rightarrow$ $Z_k = Z_k^*$ $\forall k$ vale esattamente anche in aritmetica di macchina.

$$\ker\left(H_* + I\right) = \ker\left(-J \cdot Z + I\right) = \ker\left(-J\left(Z + J\right)\right)$$

$$= \ker\left(Z + J\right).$$