Note Title 2021-03-11

crescono molto, "hump phenomenon"

$$\exp(A) = I + A + \frac{A^2}{2} + \frac{A^3}{3!} + \dots \quad \dots + \frac{A^{15}}{15!} + \dots$$

$$A = \begin{bmatrix} 0 & 30 \\ -30 & 0 \end{bmatrix} \qquad \operatorname{eig}(A) = \pm 30i \qquad \exp(A) = \begin{bmatrix} \cos(30) & \sin(30) \\ -\sin(30) & \cos(30) \end{bmatrix}$$

$\cdot\, 30i$

$\cdot\, 0$

$\cdot\, -30i$

Simile a esempi scalari:

$\exp(-30)$ no errore grande su esp. piccolo

$1/\exp(30)$ no meglio

Nel caso scalare, calcolare ad es. $\exp(-30)$ come $1/\exp(30)$ migliore cosa, ma non in questo caso.

$$\exp\left(\tfrac{1}{30}A\right)^{30} = \exp(A)$$

"scaling and squaring"

Idea: usiamo la forma di Schur $A = QTQ^*$

Q: unitaria; T: triangolare sup.

$$f(A) = Q\underbrace{f(T)}Q^*$$

$\to$ come calcolare $f(T)$, T triangolare?

$$A = \begin{bmatrix} t_{11} & t_{12} \\ 0 & t_{22} \end{bmatrix} \qquad f(A) = \begin{bmatrix} s_{11} & \boxed{s_{12}} \\ 0 & s_{22} \end{bmatrix} \qquad \begin{aligned} s_{11} &= f(t_{11}) \\ s_{22} &= f(t_{22}) \end{aligned}$$

$$Af(A) = f(A)A \qquad \begin{bmatrix} t_{11} & t_{12} \\ 0 & t_{22} \end{bmatrix}\begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix}\begin{bmatrix} t_{11} & t_{12} \\ 0 & t_{22} \end{bmatrix}$$

$$(1,2): \quad t_{11}s_{12} + t_{12}s_{22} = s_{11}t_{12} + s_{12}t_{22} \iff s_{12} = \frac{t_{12}(s_{22}-s_{11})}{t_{22}-t_{11}} = t_{12}\frac{f(t_{22})-f(t_{11})}{t_{22}-t_{11}}$$

$$t_{12}(s_{22}-s_{11}) = (t_{22}-t_{11})s_{21}$$

se $t_{11} \neq t_{22}$

converge a $t_{12} \cdot f'(t_{11})$ se $t_{22} \to t_{11}$

$$A = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ 0 & t_{22} & t_{23} \\ 0 & 0 & t_{33} \end{bmatrix} \qquad f(A) = \begin{bmatrix} s_{11} & \boxed{s_{12}} & \boxed{s_{13}} \\ 0 & s_{22} & s_{23} \\ 0 & 0 & s_{33} \end{bmatrix}$$

$$s_{12} = t_{12} \frac{s_{22} - s_{11}}{t_{22} - t_{11}}$$

$$s_{23} = t_{23} \frac{s_{33} - s_{22}}{t_{33} - t_{22}}$$

Entrata $(1,3)$ di $Af(A) = f(A)A$

$$t_{11} s_{13} + t_{12} \boxed{s_{23}} + t_{13} \boxed{s_{33}} = \boxed{s_{11}} t_{13} + \boxed{s_{12}} t_{23} + s_{13} t_{33}$$

$$s_{13} = \frac{s_{11} t_{13} + s_{12} t_{23} - t_{12} s_{23} - t_{13} s_{33}}{t_{33} - t_{11}}$$

(se $t_{11} \neq t_{33}$)

Caso generale: $Af(A) = f(A)A$, entrata (i,j) $\quad i < j$

$$t_{ii} s_{ij} + \sum_{k=i+1}^{j} t_{ik} s_{kj} = \sum_{k=i}^{j-1} s_{ik} t_{kj} + s_{ij} t_{jj}$$

Parlett recurrence $\rightsquigarrow$ $$s_{ij} = \frac{\sum_{k=i+1}^{j} t_{ik} s_{kj} - \sum_{k=i}^{j-1} s_{ik} t_{kj}}{t_{jj} - t_{ii}}$$

$\leftarrow$ entrate s_{ij} che stanno in diagonali più vicine alla principale

Posso calcolare le entrate di S in quest'ordine:

$$\begin{bmatrix} 1 & 2 & 3 & \cdots & n \\ & 1 & 2 & 3 & \vdots \\ & & 1 & \ddots & 3 \\ & 0 & & \ddots & 2 \\ & & & & 1 \end{bmatrix}$$

Stessa idea funziona a blocchi: partiziono T in blocchi (con blocchi quadrati sulla diagonale)

$$T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1k} \\ & T_{22} & \ddots & \vdots \\ 0 & & \ddots & T_{k-1,k} \\ & & & T_{kk} \end{bmatrix} \qquad S = \begin{bmatrix} S_{11} & \cdots & \cdots & S_{1k} \\ & S_{22} & & \vdots \\ 0 & & \ddots & \vdots \\ & & & S_{kk} \end{bmatrix}$$

Prendo blocco (i,j) dell'uguaglianza $Tf(T) = f(T)T$ $\quad TS = ST$

$$T_{ii} S_{ij} + \sum_{k=i+1}^{j} T_{ik} S_{kj} = \sum_{k=i}^{j-1} S_{ik} T_{kj} + S_{ij} T_{jj}$$

(*) $T_{ii} S_{ij} - S_{ij} T_{jj} = \text{noto} \Rightarrow$ eq. di Sylvester.

coefficienti già triangolari

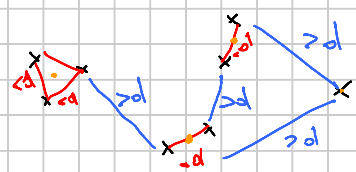
$\operatorname{sep}(T_{ii}, T_{jj}) = \sigma_{min}(I \otimes T_{ii} - T_{jj}^T \otimes I)$

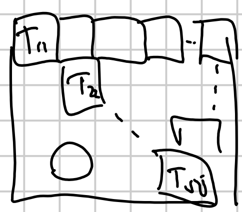
Algoritmo di Schur-Parlett

1) Calcolo $A = QTQ^*$

2) riordina la forma di Schur in modo da avere blocchi T_{ii} tali che gli autoval. di ogni T_{ii} sono "vicini" tra loro, e autovalori di blocchi diversi sono "lontani" tra loro

(es. fissando una distanza massima)



$T =$ 

3) Calcola $f(T_{ii})$ con un polinomio di Taylor centrato nella media degli autovalori $\mu = \text{media}(\text{diag}(T_{ii}))$

$$f(T_{ii}) \approx \sum_{k=0}^{d} \frac{f^{(k)}(\mu)}{k!} (T_{ii} - \mu I)^k$$

4) Calcolo i blocchi sopra la diagonale con le ricorrenze (*) (eq. di Sylvester)

5) Questo dà $f(T)$, $f(A) = Q f(T) Q^*$

Schur-Parlett con 2 blocchi:

$$T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \quad S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}$$

$S_{11} = f(T_{11})$

$S_{22} = f(T_{22})$

$TS = ST \qquad T_{11}S_{12} + T_{12}S_{22} = S_{11}T_{12} + S_{12}T_{22}$

$$\Rightarrow \boxed{T_{11}S_{12} - S_{12}T_{22} = S_{11}T_{12} - T_{12}S_{22}} \quad (**)$$ eq. Sylvester per S_{12}

Collegato a diagonalizzazione a blocchi:

$\mathrm{sep}(T_{11}, T_{22})$

$$\begin{bmatrix} I & -X \\ 0 & I \end{bmatrix}\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}\begin{bmatrix} I & X \\ 0 & I \end{bmatrix} = \begin{bmatrix} T_{11} & -XT_{22} + T_{11}X + T_{12} \\ 0 & T_{22} \end{bmatrix} = \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix}$$

Se X risolve $\boxed{T_{11}X - XT_{22} = -T_{12}}$, allora il blocco $(1,2)$ è 0

$$f\left(\begin{bmatrix} I & -X \\ 0 & I \end{bmatrix}\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}\begin{bmatrix} I & X \\ 0 & I \end{bmatrix}\right) = f\left(\begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix}\right) = \begin{bmatrix} f(T_{11}) & 0 \\ 0 & f(T_{22}) \end{bmatrix}$$

$$\Rightarrow f\left(\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}\right) = \begin{bmatrix} 1 & X \\ 0 & 1 \end{bmatrix}\begin{bmatrix} f(T_{11}) & 0 \\ 0 & f(T_{22}) \end{bmatrix}\begin{bmatrix} 1 & -X \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} f(T_{11}) & Xf(T_{22}) - f(T_{11})X \\ 0 & f(T_{22}) \end{bmatrix} \longrightarrow$$ questa quantità risolve $(**)$

$$T_{11}\left(Xf(T_{22}) - f(T_{11})X\right) - \left(Xf(T_{22}) - f(T_{11})X\right)T_{22} =$$

$$= T_{11}Xf(T_{22}) - f(T_{11})T_{11}X - XT_{22}f(T_{22}) + f(T_{11})XT_{22}$$

$$= (T_{11}X - XT_{22})f(T_{22}) - f(T_{11})(T_{11}X - XT_{22})$$

$$= -T_{12}f(T_{22}) + f(T_{11})T_{12}$$

Problemi:

1) Se prendo autoval. in blocchi diversi separati di almeno d,

allora $\lambda_{min}(I \otimes T_{11} - T_{22}^T \otimes I) \geq d$

però (se T_{ii} sono non-normali) $\sigma_{min}(I \otimes T_{11} - T_{22}^T \otimes I) \leq \lambda_{min}(\ldots)$

$\qquad \sigma_{min}(\ldots) = \mathrm{sep}(T_{11}, T_{22})$

potrebbe essere molto più piccolo

2) posso avere un singolo blocco $\Rightarrow$ costo $O(n^3\sqrt{d})$ con Paterson-Stockm.

$O(n^3 d)$ Horner o metodo naturale

3) Devo anche calcolare queste derivate!

Prossima lez: differenziazione automatica

Idea 0: derivate numeriche

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

se lavoro con precisione di macchina u, e se suppongo $f(x+h), \cancel{f}(x) \overset{f'(x)}{=} O(1)$, allora $f(x+h) \approx f(x) + f'(x)h + O(h^2)$

la differenza $f(x+h) - f(x) = f'(x)h + O(h^2)$

$O(u)$

$\frac{f(x+h)-f(x)}{h}$ ha errore dovuto alla differenza di circa $\frac{O(u)}{h}$

e errore dovuto al calcolo di $x+h$ pari a $O(u)$ $O(u)$

(TODO: sistema possibile $w(h)$)

errore da attendersi: $O(u^{1/2}) \approx 10^{-8}$

Idea alternativa: suppongo di avere una funzione specifica, ad es.

$$f(x) = \frac{x^2+5}{\exp(x)+1}$$

```
function y=f(x)
y=(x^2+5)/(exp(x)+1);
```

Trasformiamolo in codice che calcola anche funzioni di matrici:

```
function y=f(X)
n=size(X,1);
y=(X*X+5*eye(n))*inv(expm(X)+eye(n));
```

oppure $y = \mathrm{inv}(\mathrm{expm}(X) + \mathrm{eye}(n)) * (X * X + 5 * \mathrm{eye}(n))$

Se voglio calcolare la derivata $\frac{d}{dx} f$ in un punto λ, $f'(\lambda)$ mi basta chiamare

$$f\left(\begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}\right) = \begin{bmatrix} f(\lambda) & f'(\lambda) \\ 0 & f(\lambda) \end{bmatrix}$$

derivata, calcolata a meno di prec. di macchina

Per derivate più alte,

$$f\left(\begin{bmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda \end{bmatrix}\right) = \begin{bmatrix} f(\lambda) & f'(\lambda) & \frac{f''}{2}(\lambda) & \cdots & \frac{f^{(n-1)}}{(n-1)!}(\lambda) \\ & f(\lambda) & \ddots & \ddots & \vdots \\ & & \ddots & & \frac{f''(\lambda)}{2} \\ & 0 & & \ddots & f'(\lambda) \\ & & & & f(\lambda) \end{bmatrix}$$

```
function y = roba(x)
  a = 2*x^3 - 1
  z = 2/a
  while z<10
    z = 2*z
  end
  if z>12
    y = 2z
  else
    y = 3z
  end
```

```
y = roba(X)
  a = 2*X^3 - eye(n)
  z = 2*inv(a)
  while z(1,1)<10
    z = 2*z
  end
  if z(1,1) > 12
    ⋮
```