

Esercizi preliminari di calcolo matriciale

Esercizio 1 Date le seguenti matrici

$$A_1 = \begin{bmatrix} 0.4 & -1 \\ 1 & 0 \end{bmatrix}; \quad A_2 = \begin{bmatrix} 0 & 1 & 0 \\ -10 & -3 & 0 \\ 0 & 0 & -1 \end{bmatrix}; \quad A_3 = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}; \quad A_4 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},$$

determinare analiticamente

1. Le matrici inverse A_i^{-1} ;
2. I polinomi caratteristici $p_i(\lambda) = \det(\lambda I - A_i)$;
3. Gli autovalori e i relativi autovettori;
4. a) Tutte le soluzioni del sistema $A_2x = [1 \ 0 \ 0]^T$; b) Tutte le soluzioni del sistema $A_4x = 0$;
c) Tutte le soluzioni del sistema $A_4x = [1 \ 1 \ 1]^T$.

Verificare quindi i risultati ottenuti utilizzando Matlab. Comandi utili: `inv`, `pinv`, `poly`, `eig`, `det`, `roots`, `null`, `rank` (digitare `help nome-comando` per avere informazioni sulla sintassi di un comando).

Esponenziale di Matrice: L'esponenziale di una matrice quadrata è definito tramite il seguente sviluppo in serie

$$e^{At} = \sum_{k=0}^{\infty} \frac{1}{k!} A^k t^k. \quad (1)$$

È possibile dimostrare che, una volta nota la fattorizzazione di Jordan della matrice A , $A = V \Sigma V^{-1}$, dove V è la matrice contenente per colonne gli autovettori di A e Σ è la matrice di Jordan di A , l'esponenziale di matrice si può calcolare tramite la formula

$$e^{At} = V e^{\Sigma t} V^{-1}. \quad (2)$$

Per le matrici A_1, A_2, A_3, A_4 , con $t = 0.1$, calcolare i valori approssimati dell'esponenziale di matrice utilizzando lo sviluppo (1) troncato al termine $k = 1, k = 2$, etc. e confrontarli con il risultato ottenuto tramite la formula esatta (2). Confrontare i risultati ottenuti con quelli forniti dal comando Matlab `expm`.

Esercizio 2 Con l'aiuto di Matlab, calcolare il modulo e la fase delle seguenti funzioni razionali di variabile complessa, per $s = j, s = 10j, s = 100j$

$$F_1(s) = \frac{10}{s+1}; \quad F_2(s) = \frac{10}{s-1}; \quad F_3(s) = \frac{100}{s^2 + 10s + 100}.$$

Calcolare il modulo e la fase delle seguenti funzioni razionali di variabile complessa, per $z = e^j$, $s = e^{10j}$, $s = e^{100j}$

$$F_1(z) = \frac{1}{z - 0.6}; \quad F_2(z) = \frac{1}{z + 1.01}; \quad F_3(z) = \frac{.78z^{-1}}{(1 - .81z^{-1} + .606z^{-2})(1 - z^{-1})}.$$

Comandi utili: `abs`, `angle`, `polyval`, `bode`, `dbode`.

Tracce per le soluzioni

Soluzione 1 Per una matrice quadrata 2×2

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

l'inversa si calcola immediatamente come

$$A^{-1} = \frac{1}{\det A} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix},$$

dove $\det A = ad - bc$. In generale, l'inversa di una matrice A $n \times n$ si può determinare tramite la formula

$$A^{-1} = \frac{1}{\det A} \text{adj}A,$$

dove $\text{adj}A$ è usualmente detta la *matrice aggiunta* di A , ed è definita in Appendice ???. Per le matrici in esame, si ha quindi

$$A_1^{-1} = \frac{1}{1} \begin{bmatrix} 0 & 1 \\ -1 & 0.4 \end{bmatrix}.$$

Per la matrice A_2 conviene sfruttare la sua struttura detta "diagonale a blocchi" (un blocco 2×2 in alto ed un blocco 1×1 in basso). Nel caso di matrici diagonali a blocchi si ha che il determinante è pari al prodotto dei determinanti dei blocchi sulla diagonale, e l'inversa è ancora una matrice diagonale a blocchi, dove i blocchi sono dati dalle inverse dei relativi blocchi della matrice di partenza, quindi

$$A_2^{-1} = \left[\begin{array}{cc|c} -0.3 & -0.1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & -1 \end{array} \right].$$

L'inversione di A_3 non presenta problemi, mentre non esiste l'inversa di A_4 , dal momento che $\det A_4 = 0$ (matrice non invertibile).

I polinomi caratteristici si determinano calcolando il determinante della matrice $\lambda I - A$, e sono forniti da Matlab tramite il comando `poly`.

Per il calcolo degli autovalori ed autovettori, sviluppiamo nel dettaglio il caso della matrice A_4 . In questo caso il polinomio caratteristico risulta $p_4(\lambda) = \lambda(\lambda^2 - 15\lambda - 18)$, quindi gli autovalori (radici del polinomio caratteristico) risultano essere

$$\lambda_1 = 0; \quad \lambda_2 = 16.1168; \quad \lambda_3 = -1.1168.$$

Calcoliamo ad esempio l'autovettore $v^{(2)}$ relativo all'autovalore λ_2 . Tale autovettore deve risolvere il sistema di equazioni lineari

$$Av^{(2)} - \lambda_2 v^{(2)} = 0,$$

che equivale a determinare un vettore appartenente allo spazio nullo (kernel) della matrice $A - \lambda_2 I$, quindi

$$\begin{aligned} 1v_1^{(2)} + 2v_2^{(2)} + 3v_3^{(2)} - \lambda_2 v_1^{(2)} &= 0 \\ 4v_1^{(2)} + 5v_2^{(2)} + 6v_3^{(2)} - \lambda_2 v_2^{(2)} &= 0 \\ 7v_1^{(2)} + 8v_2^{(2)} + 9v_3^{(2)} - \lambda_2 v_3^{(2)} &= 0. \end{aligned}$$

Sottraendo alla seconda equazione la prima moltiplicata per 2, si ottiene $v_2^{(2)} = \frac{2(\lambda+1)}{\lambda-1}v_1^{(2)}$, che sostituito nella prima equazione fornisce $v_3^{(2)} = \frac{(\lambda-1)^2 - 4(\lambda+1)}{3(\lambda-1)}v_1^{(2)}$. Siccome una qualunque soluzione del precedente sistema omogeneo può essere moltiplicata per uno scalare arbitrario, possiamo scegliere arbitrariamente una componente del vettore (ad es. $v_1^{(2)} = 1$), e determinare le altre componenti di conseguenza:

$$v_1^{(2)} = 1; \quad v_2^{(2)} = 2.2646; \quad v_3^{(2)} = 3.5292.$$

Verificare che questa soluzione soddisfa tutte e tre le precedenti equazioni. Notare che autovalori ed autovettori possono essere ricavati direttamente tramite il comando Matlab $[V,L]=\text{eig}(A)$, che restituisce una matrice V avente per colonne gli autovettori di A ed una matrice L avente sulla diagonale i rispettivi autovalori:

```
[V,L]=eig(A)
V =
    0.2320    0.7858    0.4082
    0.5253    0.0868   -0.8165
    0.8187   -0.6123    0.4082
L =
   16.1168         0         0
         0   -1.1168         0
         0         0   -0.0000
```

Si faccia attenzione al fatto che gli autovettori restituiti da Matlab hanno in generale una normalizzazione diversa da quella utilizzata ad esempio nel calcolo manuale precedente. Verificare per esercizio la correttezza della soluzione calcolata manualmente, confrontandola con la soluzione ottenuta via Matlab.

Per quanto riguarda il Punto 4.a dell'esercizio, la matrice A_2 è invertibile, quindi la soluzione del sistema è unica e vale

$$x = A_2^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.3 \\ 1 \\ 0 \end{bmatrix}.$$

Tale soluzione può essere calcolata via Matlab nei due modi seguenti

```
A2=[0 1 0;-10 -3 0;0 0 -1];
b=[1 0 0]';
```

```
x=inv(A2)*b;
% oppure:
x=A\b
```

Per il Punto 4.b, si tratta di calcolare tutte le soluzioni (non banali) di un sistema omogeneo, cioè determinare una base dello spazio nullo di A_4 . La matrice A_4 ha un autovalore nullo, quindi la dimensione del suo spazio nullo (kernel) è pari a uno. Una base dello spazio nullo è data dall'autovettore di A_4 associato all'autovalore nullo, calcolato in precedenza

$$v^{(3)} = \begin{bmatrix} 0.4082 \\ -0.8165 \\ 0.4082 \end{bmatrix},$$

per cui tutte le possibili soluzioni del sistema sono della forma

$$x = \eta v^{(3)},$$

dove η è un qualsiasi coefficiente scalare. Una base del kernel di una generica matrice si può determinare anche tramite il comando matlab `null`.

Per il Punto 4.c, tutte le soluzioni del sistema non omogeneo $A_4 x = [1 \ 1 \ 1]^T$ possono essere espresse nella forma

$$x = A_4^\dagger b + n,$$

dove A_4^\dagger indica una *pseudo-inversa* della matrice A_4 , ed n è un qualsiasi vettore appartenente al kernel di A_4 . Una particolare pseudoinversa si calcola tramite il comando Matlab `pinv`. Per ulteriori dettagli sulla soluzione di sistemi lineari si rimanda all'Appendice ??.

Esponenziale di matrice. Dal momento che abbiamo a disposizione gli autovettori (sistemati per colonne nella matrice V) e gli autovalori (sistemati sulla diagonale della matrice Λ) delle matrici richieste, possiamo applicare direttamente la formula (2) per il calcolo esatto della matrice esponenziale. Si noti comunque che questo è fattibile facilmente solo se la matrice in questione è diagonalizzabile (cioè se Λ è una matrice diagonale), nel qual caso si ha

$$e^{\Lambda t} = \text{diag}(e^{\lambda_1 t}, e^{\lambda_2 t}, \dots, e^{\lambda_n t}).$$

Nel caso più generale (Λ composta da blocchi di Jordan J_i sulla diagonale) la formula precedente si complica, richiedendo il calcolo dell'esponenziale di ciascun blocco di Jordan che compone Λ . Riportiamo a titolo di esempio il caso di un blocco di Jordan J di ordine h :

$$J = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \lambda \end{bmatrix},$$

per il quale si ha

$$e^{Jt} = \begin{bmatrix} 1 & t & \frac{t^2}{2} & \cdots & \frac{t^{h-1}}{(h-1)!} \\ 0 & 1 & t & \cdots & \frac{t^{h-2}}{(h-2)!} \\ 0 & 0 & 1 & \cdots & \frac{t^{h-3}}{(h-3)!} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} e^{\lambda t}.$$

Tornando al problema in esame, per A_1, A_2, A_3, A_4 (che sono diagonalizzabili), con $t = 0.1$ abbiamo

$$\begin{aligned}
 e^{A_1 t} &= \begin{bmatrix} 1.0357 & -0.10186 \\ 0.10186 & 0.99494 \end{bmatrix}, \\
 e^{A_2 t} &= \begin{bmatrix} 0.95502 & 0.084963 & 0 \\ -0.84963 & 0.70013 & 0 \\ 0 & 0 & 1.1052 \end{bmatrix}, \\
 e^{A_3 t} &= \begin{bmatrix} 1.1266 & 0.21175 \\ 0.21175 & 1.0208 \end{bmatrix}, \\
 e^{A_4 t} &= \begin{bmatrix} 1.3732 & 0.53148 & 0.68981 \\ 1.0093 & 2.2481 & 1.4870 \\ 1.6454 & 1.9648 & 3.2843 \end{bmatrix}.
 \end{aligned}$$

Si osservi che questo metodo di calcolo, pur essendo formalmente “esatto,” è però numericamente mal condizionato, soprattutto nel caso di matrici non diagonalizzabili. In generale, il calcolo dell’esponenziale di matrice è un problema numericamente difficoltoso, ed anche il metodo dello sviluppo in serie (1) non da risultati particolarmente soddisfacenti, specie se troncato al primo ordine. Il seguente codice Matlab calcola lo sviluppo della serie (1), troncato ad un ordine N prefissato, e valuta l’errore rispetto alla soluzione “esatta” calcolata da `expm`.

```

A=[1 2 3;4 5 6;7 8 9];
t=0.1;
% Esponenziale "esatto":
Em=expm(A*t);
% Calcolo approssimato tramite sviluppo in serie
N=10; % ordine dello sviluppo
%
n=size(A,1);
E=eye(n);
kf=1;
At_k=eye(n);
for k=1:N
    At_k=At_k*A*t;
    E=E+At_k/kf;
    kf=kf*(kf+1);
end;
% la matrice esponenziale approssimata e' contenuta in E.
% Calcolo errore di approssimazione:
err=norm(Em-E);
disp(['Errore di approssimazione relativo: ',...
      num2str(err/norm(Em)*100),' %']);

```

Soluzione 2 Per il modulo di una funzione razionale $F(s) = \frac{N(s)}{D(s)}$, dove $N(s)$ e $D(s)$ sono

polinomi nella variabile complessa s , si ha

$$|F(s)| = \frac{|N(s)|}{|D(s)|}.$$

Il modulo di $N(s)$ e $D(s)$ si calcola applicando l'algebra elementare dei numeri complessi, i.e. $|a + jb| = \sqrt{a^2 + b^2}$ (con a, b reali), $j^2 = -1$.

Per la fase di $F(s)$ (che si denota con $\arg F$) si ha

$$\arg F(s) = \arg N(s) - \arg D(s).$$

La fase di $N(s)$ e $D(s)$ si calcola applicando l'algebra elementare dei numeri complessi, i.e. $\arg(a + jb)^n = n \arctan b/a$ (con a, b reali). Per ulteriori dettagli sulle operazioni con i numeri complessi si veda l'Appendice ??.

Nel caso in esame, per $F_1(s)$, $s = j\omega$ si ha

$$|F_1(j\omega)| = \frac{10}{\sqrt{1 + \omega^2}} = \begin{cases} 7.0711, & \text{per } \omega = 1, \\ 0.995, & \text{per } \omega = 10, \\ 0.1, & \text{per } \omega = 100; \end{cases}$$

$$\arg F_1(j\omega) = -\arctan \omega = \begin{cases} -45^\circ, & \text{per } \omega = 1, \\ -84.2894^\circ, & \text{per } \omega = 10, \\ -89.4271^\circ, & \text{per } \omega = 100. \end{cases}$$

Modulo e fase di $F(s)$, per $s = j\omega$, si possono calcolare tramite Matlab, utilizzando il comando `[modulo, fase]=bode(num, den, omega)`, dove `num` e `den` sono vettori che rappresentano i polinomi a numeratore e a denominatore, rispettivamente.

Per le funzioni della variabile $z = e^{j\omega}$ procediamo in modo analogo

$$|F_1(e^{j\omega})| = \frac{1}{|e^{j\omega} - 0.6|} = \frac{1}{|\cos \omega + j \sin \omega - 0.6|} =$$

$$\frac{1}{\sqrt{(\cos \omega - 0.6)^2 + \sin^2 \omega}} = \begin{cases} 1.1854, & \text{per } \omega = 1, \\ 0.6500, & \text{per } \omega = 10, \\ 1.7535, & \text{per } \omega = 100; \end{cases}$$

$$\arg F_1(e^{j\omega}) = -\arctan \frac{\sin \omega}{\cos \omega - 0.6} = \begin{cases} -94.0580^\circ, & \text{per } \omega = 1, \\ 159.2916^\circ, & \text{per } \omega = 10, \\ 62.6139^\circ, & \text{per } \omega = 100. \end{cases}$$

Per la funzione $F_3(z)$ è consigliabile dapprima moltiplicare numeratore e denominatore per z^3 , in modo da riportarsi al rapporto tra due polinomi in z . Nel caso di funzioni in $z = e^{j\omega}$ si può utilizzare il comando Matlab

$$[\text{modulo}, \text{fase}] = \text{dbode}(\text{num}, \text{den}, 1, \text{omega}),$$

dove `num` e `den` sono vettori che rappresentano i polinomi in z a numeratore e a denominatore, rispettivamente.