Anna Dellepiane (5565836), Marco Zoratti (5562866).

Relazione Laboratorio 3 MATLAB

Esercizio 1

L'esercizio 1 è stato sviluppato nel file *esercizio1.m* .

All'interno dello script, è stata creata la matrice A di dimensione nxn con $\mathrm{n} = 76$ dovuto ad un calcolo effettuato con le ultime due cifre del codice matricola di Zoratti Marco (5562866).

È stata successivamente creata la matrice perturbata B, pari ad A con l'elemento B(n,1) perturbato, avente come valore 2^{-n}.

Il punto a è stato svolto calcolando tramite il comando *eig* gli autovalori delle matrici A e B, inseriti rispettivamente nelle variabili *autovalori_A* e *autovalori_B*; successivamente sono stati calcolati errore relativo in input ed errore relativo in output.

Per lo svolgimento del punto b è stato ripetuto quanto scritto sopra per le matrici A^TA e B^TB.

Risultati ottenuti

Punto a

autovalori_A	[1, 1, … 1]
autovalori_B	[1, 1, … 1]
errIn_a	6.618840155214590e-24
errOut_b	0

Per entrambe le matrici è presente un unico autovalore λ = 1 con molteplicità n, ovvero 76 nel nostro caso.

L'errore relativo in input rilevato è pari a 6.618840155214590e-24, ovvero $\frac{2^{-n}}{2}$.

L'errore relativo in output invece è pari a 0 dato che *autovalori_A* e *autovalori_B* sono uguali.

Punto b

errIn_b	3.310117852093688e-24
errOut_b	0

L'errore relativo in output è sempre pari a 0 dato che gli autovalori delle due matrici sono coincidenti, mentre quello in input è pari a 3.310117852093688e-24. Si noti come questo sia minore dell'errore relativo in input del punto a.

Esercizio 2

Come per il primo esercizio, l'esercizio 2 è stato sviluppato nel file *esercizio2.m* .

Per lo svolgimento del primo punto è stata creata la matrice A di dimensioni 11x11, ovvero la matrice di adiacenza per il grafo dei collegamenti ferroviari della regione Lombardia. Essendo un grafo non orientato, i collegamenti sono simmetrici e quindi anche la matrice.

È stato poi realizzato il punto b, creando una matrice D 11x11 diagonale, tale che $\forall g_j$ con g_j elemento della diagonale, g_j = numero archi uscenti dal nodo j. Questa matrice è stata realizzata tramite la funzione *diag* e la funzione *sum:* la funzione *sum* ritorna un vettore contenente per ogni colonna j il numero di archi entranti in j. Questo vettore risultato viene dato in input alla funzione *diag*, che ritorna la matrice D avente nella diagonale il numero di archi uscenti per ogni nodo j.

Successivamente è stata calcolata la matrice G = $A \cdot D^{-1}$, di modo da poterne calcolare la matrice degli autovettori e la matrice diagonale degli autovalori, chiamate rispettivamente *autovettori* e *autovalori*.

Nota: Questi ultimi riportavano numeri complessi quindi è stata applicata la funzione *real* di modo da avere solamente componenti reali e non immaginarie.

Per il punto c, si faccia riferimento ai risultati ottenuti sotto.

Risultati ottenuti

Autovalori di G (guardare i valori diagonali di *autovalori*):

	1	2	3	4	5	6	7	8	9	10	11
1	1.0000	0	0	0	0	0	0	0	0	0	0
2	0	0.7640	0	0	0	0	0	0	0	0	0
3	0	0	0.5774	0	0	0	0	0	0	0	0
4	0	0	0	0.2241	0	0	0	0	0	0	0
5	0	0	0	0	-0.8824	0	0	0	0	0	0
6	0	0	0	0	0	-0.7255	0	0	0	0	0
7	0	0	0	0	0	0	-0.5774	0	0	0	0
8	0	0	0	0	0	0	0	-0.3802	0	0	0
9	0	0	0	0	0	0	0	0	-2.1440e-17	0	0
10	0	0	0	0	0	0	0	0	0	-2.1440e-17	0
11	0	0	0	0	0	0	0	0	0	0	5.3307e-17

Si noti che solo il primo elemento ha il modulo di valore 1, mentre i successivi sono minori.

Autovettori di G:

	1	2	3	4	5	6	7	8	9	10	11
1	0.6124	-0.0521	-0.7385	-0.2314	0.6272	0.3765	0.7385	-0.0708	-9.8846e-17	-9.8846e-17	-4.1617e-16
2	0.1021	-0.0114	-0.2132	-0.1721	-0.1185	-0.0865	-0.2132	0.0310	-0.2424	-0.2424	0.4589
3	0.2041	-0.2634	2.4610e-17	-0.2283	-0.3212	0.0441	-3.4949e-16	0.1304	0.0465	0.0465	0.4353
4	0.3062	-0.1727	1.5996e-16	0.4857	-0.3307	0.1606	-1.1941e-16	-0.3644	0.5008	0.5008	0.1766
5	0.4082	0.2772	2.4610e-16	0.6400	0.0335	-0.3382	-3.3784e-16	0.7525	-9.7850e-17	-9.7850e-17	-1.5436e-16
6	0.3062	0.3086	1.2305e-16	0.2539	-0.1622	-0.2735	-3.7279e-16	-0.5011	-0.5008	-0.5008	-0.1766
7	0.1021	-0.0114	-0.2132	-0.1721	-0.1185	-0.0865	-0.2132	0.0310	0.2191	0.2191	-0.6765
8	0.3062	0.5255	0.3693	-0.1936	0.0905	0.6606	-0.3693	0.0425	-6.4960e-17	-6.4960e-17	-1.0612e-16
9	0.1021	0.2293	0.2132	-0.2880	-0.0342	-0.3036	0.2132	-0.0373	0.1669	0.1669	0.0589
10	0.3062	-0.5776	0.3693	-0.0378	0.5367	-0.2842	-0.3693	-0.1133	1.7146e-17	1.7146e-17	2.0777e-16
11	0.1021	-0.2520	0.2132	-0.0562	-0.2028	0.1306	0.2132	0.0994	-0.1902	-0.1902	-0.2765

Analizzando la matrice degli autovettori, si può notare che l'autovettore relativo all'autovalore 1 presenta componenti tutte comprese tra 0 e –1; è stato quindi moltiplicato per –1 di modo da ottenere tutte componenti comprese tra 0 e 1.

Infine, gli autovettori corrispondenti agli altri autovalori hanno componenti sia positive sia negative.

“Importanza” delle stazioni ferroviarie

Componente m	**Città**	**Numero di collegamenti**
0.6124	Milano	6
0.4082	Bergamo	4
0.3062	Brescia	3
0.3062	Como	3
0.3062	Lecco	3
0.3062	Cremona	3
0.2041	Lodi	2
0.1021	Pavia	1
0.1021	Varese	1
0.1021	Sondrio	1
0.1021	Mantova	1

Per concludere, dalla tabella sopra si può notare come le componenti *m* del vettore *x* attribuiscano un valore di “importanza” a ciascuna città presente nel grafo; si può dedurre che le città aventi un numero maggiore di collegamenti sono anche quelle più importanti all’interno del grafo.

Esercizio 3

A differenza dei primi due esercizi, per il terzo sono stati sviluppati più file contenenti funzioni ausiliarie per migliorare la leggibilità del codice e il suo riutilizzo, in particolare:

- ***metodoPotenze.m***;
  Il file contiene una funzione omonima che presa una matrice A e un vettore in input v, esegue il metodo delle potenze su A con v come vettore iniziale.
- ***metodoPotenzeInv.m***;
  All’interno del file è contenuta una funzione omonima che data una matrice A, un vettore v e un numero p in input, esegue il metodo delle potenze inverso su A con v come vettore iniziale e p come valore dello shift.
- ***velocitaConvergenza.m***;
  Il file contiene la funzione omonima che dato il vettore degli autovalori in input, calcola la velocità di convergenza.

Lo script *esercizio3.m* invece crea la matrice A richiesta, i vettori v1 e v2 ed esegue il metodo delle potenze (grazie a *metodoPotenze*) su A con questi ultimi come vettori iniziali. Tramite *eig* vengono poi calcolati gli autovalori di A.

Viene successivamente eseguito il punto b, definendo un valore per lo shift ed eseguendo il metodo delle potenze inverse su A sempre con i vettori v1 e v2, usando la funzione *metodoPotenzeInv*. Come valore dello shift è stato scelto il valore 6, questo perché l’autovalore di massimo modulo è pari a 5.

Risultati ottenuti

lmbd1	[9.69, 7.45, 5.77, 5.24, 5.14, … 5.0000]
lmbd2	[18.46, 3.74, 2.79, 3.07, 2.97, … 3.0000]
lmbd3	[5.39, 5.17, 5.06, 5.02, … 5.0000]
lmbd4	[2.20, 2.63, 2.84, 2.97, 2.99, … 3.0000, …, 5.0000]

Nota: i valori riportati nella tabella sono un'approssimazione dei valori reali calcolati su Matlab dovuta a motivi di leggibilità.

Nota 2: *lmbd1* e *lmbd3* fanno riferimento al vettore v1 mentre *lmbd2* e *lmbd4* fanno riferimento al vettore v2.

Osservando la tabella si può notare come il metodo delle potenze inverso approssimi "più velocemente" all'autovalore di massimo modulo rispetto al metodo delle potenze.

Sono state infine calcolate le velocità di convergenza dei due metodi: per il metodo delle potenze è pari a $0.6^k = \left(\frac{3}{5}\right)^k$ mentre per il metodo delle potenze inverso abbiamo una velocità di convergenza pari a $0.3333^k = \left(\frac{1}{3}\right)^k$.

Ambiente di sviluppo e altro

L'intero laboratorio è stato sviluppato su MATLAB aggiornato alla versione:

23.2.0.2459199 (R2023b) Update 5.