Esercitazione 4

Argomento: autovalori e valori singolari

1. Utilizzare il comando `eig` per calcolare gli autovalori delle matrici

$$\mathbf{A}_1 = \begin{pmatrix} n & n-1 & n-2 & \cdots & 3 & 2 & 1 \\ n-1 & n-1 & n-2 & \cdots & 3 & 2 & 1 \\ 0 & n-2 & n-2 & \cdots & 3 & 2 & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & \vdots & \ddots & \ddots & 3 & 2 & 1 \\ 0 & \vdots & \ddots & \ddots & 2 & 2 & 1 \\ 0 & 0 & \cdots & \cdots & 0 & 1 & 1 \end{pmatrix}, \ \mathbf{A}_2 = \begin{pmatrix} n & n-1 & n-2 & \cdots & 3 & 2 & 1 \\ n-1 & n-1 & n-2 & \cdots & 3 & 2 & 1 \\ n-2 & n-2 & n-2 & \cdots & 3 & 2 & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 3 & \vdots & \ddots & \ddots & 3 & 2 & 1 \\ 2 & \vdots & \ddots & \ddots & 2 & 2 & 1 \\ 1 & 1 & \cdots & \cdots & 1 & 1 & 1 \end{pmatrix},$$

   con $n = 100$.

   Successivamente introdurre una perturbazione $\varepsilon = 1.0e-10$ negli elementi dell'ultima riga di entrambe le matrici, e calcolare gli autovalori delle matrici perturbate.

   Rappresentare graficamente gli autovalori e giustificare i risultati ottenuti.

2. Implementare il metodo delle potenze in una *function* con nome `potenze.m`; fissare un numero massimo `m_max` di iterazioni da eseguire e una precisione relativa `toll`.

   Successivamente, a partire dal vettore $\mathbf{z} = (1, 2, 3)^T$, determinare con una tolleranza pari a `toll=1.0e-10` l'autovalore di modulo massimo delle seguenti matrici:

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 2 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \ \mathbf{A}_2 = \begin{pmatrix} 0.1 & 3.8 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \ \mathbf{A}_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

   Giustificare i risultati utilizzando il comando `eig` di MATLAB.

3. In una *function* denominata `potenze_no_norma.m`, implementare il metodo delle potenze che non prevede la normalizzazione del vettore iterata a ogni passo.

   Successivamente, a partire dal vettore $\mathbf{z} = (1, 2, 3)^T$, determinare l'autovalore di modulo massimo e un autovettore ad esso associato della matrice $\mathbf{A}_2$ dell'esercizio precedente. Fissare `m_max`=500,800,1100 e commentare i risultati.

4. In una *function* denominata `potenze_inverse.m`, modificare opportunamente l'algoritmo implementato in `potenze.m` in modo tale da ottenere l'algoritmo che implementa il metodo delle potenze inverse.

   Successivamente, applicare la *function* `potenze_inverse.m` per determinare l'autovalore più vicino a $p = 0.5$, con precisione relativa $toll = 1.0e-10$, delle seguenti matrici:

$$\mathbf{A}_1 = \begin{pmatrix} 1 & -2 & 0 \\ 0 & 2 & 0 \\ 1 & 1 & 3 \end{pmatrix}, \ \mathbf{A}_2 = \begin{pmatrix} 0.5 & -2 & 0 \\ 0 & 2 & 0 \\ 1 & 1 & 3 \end{pmatrix}, \ \mathbf{A}_3 = \begin{pmatrix} 0 & -2 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 3 \end{pmatrix}.$$

   Richiamare la *function* `eigs.m` di MATLAB e, alla luce dei valori ottenuti, commentare i risultati prodotti da `potenze_inverse.m`.

5. Utilizzando le proprietà dei cerchi di Gershgorin, determinare un opportuno p per il calcolo dell'autovalore di modulo massimo della seguente matrice:

$$\mathbf{A} = \begin{pmatrix} 5 & 0 & 1 & -1 \\ 0 & 2 & 0 & -\frac{1}{2} \\ 0 & 1 & -1 & 1 \\ -1 & -1 & 0 & 0 \end{pmatrix}.$$

   Confrontare il numero di iterazioni e il costo computazionale del metodo delle potenze e del metodo delle potenze inverse richiedendo una tolleranza pari a 10^{-14}.

6. Implementare il metodo QR, nella sua forma più elementare, in una *function* con nome `qr_base.m`; fissare un numero massimo `m_max` di iterazioni e il seguente criterio d'arresto `norm(tril(A,-1),inf) <= toll`, ove `toll` è un parametro di input.

   Applicare il suddetto metodo, alla matrice di Hilbert di ordine 10, ponendo `m_max=100` e `toll=1.0e-14`.

   Calcolare l'errore associato ai suddetti autovalori, prendendo come valori di riferimento quelli ottenuti con il comando `eig`.

   Successivamente, eseguire 100 iterazioni del metodo QR applicato alla matrice

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 2 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

   Commentare i risultati.

7. Selezionare il formato `format long e`. Definire le matrici triangolari superiore $\mathbf{A}$ di ordine $n = 5, 10, 15, ..., 100$, i cui elementi sono

$$a_{ij} = \begin{cases} 1 & \text{se } i = j \\ -1 & \text{se } i < j \\ 0 & \text{se } i > j \end{cases}$$

   Calcolare il determinante, il rango e i valori singolari delle matrici $\mathbf{A}$ e commentare i risultati.

8. Sia $\mathbf{Ax} = \mathbf{b}$ con $\mathbf{A}$ matrice di Hilbert di ordine $n = 15$ e $\mathbf{b}$ definito in modo tale che la soluzione esatta $\mathbf{x}$ del sistema coincida con il vettore unitario.

   Risolvere il suddetto sistema con il comando MATLAB `x=A\b` e con la decomposizione ai valori singolari, dopo aver approssimato la matrice $\mathbf{A}$ con una matrice di rango inferiore ottenuta trascurando i valori singolari al di sotto della tolleranza $tol = 10^{-14}$.

   Confrontare gli errori relativi associati alla soluzione ottenuta con i due metodi.

9. Si consideri il sistema lineare

$$\begin{aligned} 3x_1 - 2x_2 + x_3 + 2x_4 &= 1 \\ -x_1 + 2x_3 + x_4 &= -3 \\ 5x_2 - 6x_3 - x_4 &= 7 \\ x_1 + x_2 - x3 + x_4 &= 0 \\ x_1 - x_2 - x_3 - x_4 &= -6 \\ 8x_1 - x_2 - 5x_3 + 2x_4 &= 2 \end{aligned}$$

   Calcolare il rango della matrice A del sistema e, successivamente, calcolare la soluzione del sistema nel senso dei minimi quadrati, avente norma euclidea minima.

   Verificare la correttezza del risultato utilizzando il comando `pinv` di Matlab.

   Quali sono tutte le soluzioni nel senso dei minimi quadrati del sistema?

10. Utilizzare la decomposizione SVD per approssimare l'immagine cucciolo.jpg, generata dalla *blackness matrix* di dimensione 271×300, mediante una matrice di rango $n = 10, 30, 50, 70$.