CORSO DI CALCOLO NUMERICO (3 CREDITI)

DOCENTE: L. SCUDERI

ESERCITAZIONE 2

Argomenti: sistemi lineari

N.B. Tutte le *function* richieste o citate che non sono di MATLAB sono disponibili all'indirizzo: `http://calvino.polito.it/~scuderi/programmi_libro`.

1. Generare una matrice A di ordine n e un vettore b di dimensione $n \times 1$, con $n = 1000, 2000$, di numeri pseudo-casuali. Utilizzare i comandi `tic` e `toc` di MATLAB per calcolare il tempo d'esecuzione richiesto dagli algoritmi `x=A\b` e `x=inv(A)*b` per la risoluzione del sistema lineare $Ax = b$. Commentare i risultati.

2. Selezionare il formato di output `format long e` e risolvere i sistemi lineari $Ax = b$ con A matrice di Hilbert di ordine $n = 5, 10, 15$ e b definito in modo tale il corrispondente vettore soluzione x coincida con il vettore unitario. Per ciascun valore di n, calcolare l'errore relativo della soluzione e il numero di condizionamento. Commentare i risultati.

3. Implementare in due *m-file* di tipo *function*, denominati `forward.m` e `backward.m`, il metodo di sostituzione in avanti e quello all'indietro per la risoluzione dei sistemi lineari $Lx = b$ e $Ux = b$ associati ad una matrice triangolare inferiore e superiore, rispettivamente.

   Strutturare le *function* `forward` e `backward` in modo tale che, ricevendo in input la matrice dei coefficienti e il vettore dei termini noti, restituiscano in output il vettore soluzione ed una variabile *ier* che informa l'utente sulla eventuale singolarità della matrice dei coefficienti (in particolare *ier* vale 0 se la matrice è singolare, vale 1 altrimenti).

   Successivamente risolvere i sistemi lineari $Lx = b$ e $Ux = b$ con L ed U di ordine $n = 10$ e di forma triangolare inferiore e superiore rispettivamente con tutti gli elementi non nulli uguali ad 1, tranne quelli della diagonale principale che sono uguali a 2. Definire b in modo tale che la soluzione x dei sistemi lineari corrispondenti coincida con il vettore unitario.

4. Generare la matrice A di ordine $n = 18$, che ha per elementi $a_{ij} = \cos((j-1)\theta_i)$, $\theta_i = (2i-1)\pi/(2n)$, $i, j = 1, \ldots, n$, e definire il vettore b in modo tale che la soluzione del sistema $Ax = b$ coincida col vettore unitario. Quindi selezionare il formato di output `format long e` e risolvere il sistema $Ax = b$ col metodo di eliminazione di Gauss senza pivoting mediante la *function* `gaussnopiv` e con pivoting parziale mediante l'istruzione `x=A\b` di MATLAB. Stampare il condizionamento in norma infinito della matrice A, gli errori relativi in norma infinito delle soluzioni ottenute mediante i due algoritmi e commentare i risultati.

---

5. Generare la seguente matrice

$$A = \begin{pmatrix} 11 & 2 & 3 & 4 \\ 0 & 8 & 2 & 3 \\ 0 & 0 & 4 & 0 \\ 1 & 0 & 0 & 5 \end{pmatrix}.$$

Determinare le matrici P, L e U della fattorizzazione $PA = LU$ della matrice A mediante la *function* `lu` di MATLAB. Successivamente utilizzare i suddetti fattori per risolvere il sistema lineare $Ax = b$, con b definito in modo tale che la corrispondente soluzione x coincida con il vettore unitario. Esiste la decomposizione di Choleski della matrice A? Perché? Giustificare il risultato che si ottiene utilizzando il comando `chol` di MATLAB.

6. Risolvere in modo efficiente, minimizzando il numero di operazioni aritmetiche, i seguenti sistemi
$$\begin{cases} Ax_1 = b_1 \\ Ax_2 = b_2 \\ Ax_3 = b_3 \\ Ax_4 = b_4 \end{cases}$$
aventi tutti la stessa matrice dei coefficienti A, non singolare, di ordine 6 e costituita da numeri pseudo-casuali; inoltre il termine noto b_1 è definito in modo tale che la corrispondente soluzione x coincida con il vettore unitario e $b_i = b_{i-1}/i$, $i = 2, 3, 4$.

7. Risolvere in modo efficiente il sistema
$$A^4 z = b$$
ove A è non singolare, di ordine 5 e costituita da numeri pseudo-casuali e b è definito in modo tale che la corrispondente soluzione z coincida con il vettore unitario.

8. Generare una matrice A di numeri pseudo-casuali, non singolare e di ordine 3, e determinare le matrici P, L e U della fattorizzazione $PA = LU$ della matrice A mediante la *function* `lu` di MATLAB. Successivamente utilizzare i suddetti fattori per invertire la matrice A. Utilizzare la *function* `inv` di MATLAB per verificare la correttezza del risultato.

9. Generare una matrice B di numeri pseudo-casuali, non singolare e di ordine 5. Verificare che la matrice $A = B^T B$ è simmetrica e definita positiva ed utilizzare la *function* `chol` di MATLAB per determinare la decomposizione di Choleski $A = L_1 L_1^T$.

   Successivamente utilizzare la suddetta decomposizione per calcolare la matrice inversa di A e per risolvere il sistema lineare $Ax = b$, con b definito in modo tale che la corrispondente soluzione x coincida con il vettore unitario.

10. Utilizzare le *function* `jacobi.m` e `gauss_seidel.m`, in cui sono stati implementati i metodi iterativi di Jacobi e di Gauss-Seidel rispettivamente, per la risoluzione dei sistemi $A_i x = b_i$, $i = 1, \ldots, 4$, con
$$A_1 = \begin{pmatrix} 1 & -2 & 2 \\ -1 & 1 & -1 \\ -2 & -2 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 4 & 0 & 2/5 \\ 0 & 5 & 2/5 \\ 5/2 & 2 & 1 \end{pmatrix},$$
$$A_3 = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 3 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & 3 \end{pmatrix},$$
e b_i definito in modo tale che la corrispondente soluzione x coincida con il vettore unitario. Per ciascuno di essi richiedere la tolleranza relativa $toll = 1.0\text{e}-7$, un numero massimo di iterazioni $itermax = 100$ e fornire il vettore nullo come vettore iniziale x_0. Commentare i risultati.