Metodi Numerici per il Calcolo

Esercitazione 8: $A\mathbf{x} = \mathbf{b}$ e Fattorizzazione LU

A.A.2022/23

Scaricare dalla pagina web del corso l'archivio matlab_mnc2223_8.zip e scompattarlo nella propria home directory. Verrà creata una cartella con lo stesso nome contenente alcuni semplici script e function Matlab/Octave. Si svolga la seguente esercitazione che ha come obiettivo sperimentare la fattorizzazione LU di una matrice e la soluzione di sistemi lineari.

Nella cartella sono presenti alcune function che se richiamate restituiscono una matrice $n \times n$ non singolare che può essere utilizzata come matrice test per un sistema lineare $A\mathbf{x} = \mathbf{b}$. Come metodologia di lavoro si proceda definendo il vettore $\mathbf{x} \in \mathbb{R}^n$ (per esempio $\mathbf{x} = (1, 1, \ldots, 1)^T$) e si determini $\mathbf{b}$ affinché la soluzione del sistema sia il vettore $\mathbf{x}$, così da conoscerne la soluzione esatta.

A. Function `lu` di Matlab/Octave

Completare lo script/function `main_linsys.m` che definito un sistema lineare $A\mathbf{x} = \mathbf{b}$, lo risolve nei due seguenti modi:

- utilizzando l'operatore "left-division" di Matlab/Octave;
- utilizzando la function di Matlab/Octave `lu` che implementa la fattorizzazione di Gauss con scambio delle righe e perno massimo. Più precisamente:

  ```
  [L,U]=lu(A)
  [L,U,P]=lu(A)
  ```

  (vedere `help lu`) quindi si usino le function `lsolve.m` e `usolve.m` presenti nella cartella per risolvere i sistemi

  $$L\mathbf{y} = P\mathbf{b}$$
  $$U\mathbf{x} = \mathbf{y}.$$

B. Sulla fattorizzazione LU

Le function `LUsimple.m` e `LUmaxpiv.m` implementano rispettivamente la fattorizzazione LU di Gauss semplice (senza scambio di righe) e con scambio di righe e perno massimo.

- Analizzare i loro codici e confrontarli con quanto visto a lezione.
- Completare lo script `main_gauss_simple.m` per sperimentare l'algoritmo di Gauss semplice sulle matrici test presenti nella cartella e per risolvere i relativi sistemi lineari (si utilizzino le function `LUsimple.m` e `LUsimple_solve.m`).

- Modificare lo script (lo si chiami `main_gauss_maxpiv.m`) per sperimentare l'algoritmo di Gauss con scambio delle righe e perno massimo (si utilizzino le function `LUmaxpiv.m` e `LUmaxpiv_solve.m`).

C. Sulla stabilità della fattorizzazione LU

Si completi lo script `main_lufact.m` in modo che richiami la function `lu` di Matlab come nell'esercizio A (fattorizzazione LU di una matrice con scambio delle righe e perno massimo) e verifichi che:

$$\max|\ell_{i,j}| \leq 1, \qquad \max|u_{i,j}| \leq 2^{n-1}\max|a_{i,j}|$$

dove $\ell_{i,j}$ e $u_{i,j}$ sono gli elementi delle matrici L e U determinate. Si stampi una tabella con i seguenti valori per le matrici di esempio `mat_k, k=2,3,4,5` di dimensioni $n = 5, 10, 50$

mat_k	$n \times n$	$\max\|\ell_{i,j}\|$	$\max\|u_{i,j}\|$	$2^{n-1}\max\|a_{i,j}\|$

D. Sul condizionamento di un sistema lineare

Si consideri la matrice H di Hilbert $n \times n$ (function Matlab `hilb`, vedi l'`help`) Si calcoli $\mathbf{b}$ in modo che $H\mathbf{x} = \mathbf{b}$, dove $\mathbf{x} = (1, 1, \ldots, 1)^T$. Si aggiunga a $\mathbf{b}$ una perturbazione

$$\delta\mathbf{b}_p = 10^{-p}\texttt{rand}(n, 1).$$

Si risolva il sistema $H\tilde{\mathbf{x}}_p = \mathbf{b} + \delta\mathbf{b}_p$, per $p = 1, \ldots, 5$ e si stampi per ogni valore di p la seguente quantità:

$$K_p = \frac{\|\mathbf{x} - \tilde{\mathbf{x}}_p\|}{\|\mathbf{x}\|}\frac{\|\mathbf{b}\|}{\|\delta\mathbf{b}_p\|}.$$

I valori K_p così ottenuti sono il valore effettivo sperimentale del numero di condizione della matrice H. Verificare che per ogni p sia

$$K_p \leq \texttt{cond}(H).$$

Lo script `main_hilb.m` implementa già quanto detto; eseguire per differenti dimensioni $n \times n$ e analizzare i risultati.

E. Sul condizionamento delle matrici usate per l'interpolazione

Nella cartella sono presenti le function `vandermonde.m`, `newton.m` e `bernst.m` già utilizzate nell'interpolazione polinomiale e che generano, avendo definito un vettore di punti da interpolare, le matrici del sistema lineare relativo:

- base canonica (matrice di Vandermonde);
- forma di Newton (matrice triangolare);

- base di Bernstein (matrice stocastica).

Si completi la function `main_interp.m` e si utilizzi la function Matlab/Octave `cond` per avere una stima del numero di condizione di tali matrici. Si confrontino i numeri di condizione ottenuti per le matrici suddette di dimensioni `n=5:5:20` sia per punti equispaziati che di Chebyshev.

F. Esercizio di verifica (Stabilità della fattorizzazione LU)
Con riferimento alla function `LUsimple.m` e `LUmaxpiv.m` del precedente esercizio B, si completi lo script `main_stab_lufact.m` per risolve il seguente sistema lineare $A\mathbf{x} = \mathbf{b}$ con

$$A = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

utilizzando i due algoritmi. Si confrontino le soluzioni.