

Risoluzione numerica di sistemi lineari

1. METODI DIRETTI

Risoluzione di sistemi lineari con matrice generica. Scrivere uno script Python che:

- (a) crea un problema test di dimensione variabile n la cui soluzione esatta sia il vettore x di tutti elementi unitari e b il termine noto ottenuto moltiplicando la matrice A per la soluzione x .
- (b) calcola il numero di condizione (o una stima di esso)
- (c) risolve il sistema lineare $Ax = b$ con la fattorizzazione LU con pivoting.

Problemi test

- Una matrice di numeri casuali A generata con la funzione `randn` di Matlab, (n variabile fra 10 e 1000)

Risoluzione di sistemi lineari con matrice simmetrica e definita positiva.

Scrivere uno script Python che:

- (a) crea un problema test di dimensione variabile n la cui soluzione esatta sia il vettore x di tutti elementi unitari e b il termine noto ottenuto moltiplicando la matrice A per la soluzione x .
- (b) calcola il numero di condizione (o una stima di esso)
- (c) risolve il sistema lineare $Ax = b$ con la fattorizzazione di Cholesky..

Problemi test

- matrice di Hilbert di dimensione n (con n variabile fra 2 e 15)
- matrice tridiagonale simmetrica e definita positiva avente sulla diagonale elementi uguali a 9 e quelli sopra e sottodiagonali uguali a -4.

Per ogni problema test:

- Disegna il grafico del numero di condizione in funzione della dimensione del sistema
- Disegna il grafico dell'errore in norma 2 in funzione della dimensione del sistema

2. METODI ITERATIVI

Scrivi le funzioni `Jacobi(A, b, x0, maxit, tol, xTrue)` e `GaussSeidel(A, b, x0, maxit, tol, xTrue)` per implementare i metodi di Jacobi e di Gauss Seidel per la risoluzione di sistemi lineari con matrice a diagonale dominante. In particolare:

- `x0` sia l'iterato iniziale;
- la condizione d'arresto sia dettata dal numero massimo di iterazioni consentite `maxit` e dalla tolleranza `tol` sulla differenza relativa fra due iterati successivi;
- si preveda in input la soluzione esatta `xTrue` per calcolare l'errore relativo ad ogni iterazione.

Entrambe le funzioni restituiscano in output:

- la soluzione \mathbf{x} ;
- il numero k di iterazioni effettuate;
- il vettore `relErr` di tutti gli errori relativi.

Test

- considerare la precedente matrice tridiagonale per $N \leq 100$.
- verificare, calcolando il raggio spettrale della matrice, la convergenza dei metodi.
- Eseguire i metodi con tolleranza $1.e-8$.
- Riportare in un grafico l'errore relativo (in norma 2) di entrambi i metodi al variare del numero di iterazioni per N fissato (scegliere almeno due valori di N).
- Riportare in un grafico l'errore relativo finale dei metodi al variare della dimensione N del sistema.
- riportare in un grafico il numero di iterazioni di entrambi i metodi al variare di N
- Riportare in un grafico il tempo impiegato dai metodi di Jacobi, Gauss Sidel, LU, Cholesky al variare di N .

Traccia per la discussione

1. Utilizzando i grafici richiesti: spiegare l'andamento dell'errore rispetto al numero di condizione della matrice, l'andamento del tempo di esecuzione rispetto alla dimensione del sistema in relazione alla complessità computazionale degli algoritmi utilizzati.
2. Discutere la differenza di errore e tempo di esecuzione ottenuti con i metodi diretti e iterativi.