

Corso di Laurea in Ingegneria Biomedica

Laboratorio di Algebra Lineare Numerica

A.A. 2019/2020 – I Ciclo

Esercitazione 3

Creare una cartella <cognome> in C: dove verranno salvati i file creati nella sessione di lavoro. Appena entrati in MATLAB posizionarsi in <cognome>.
Risolvere in ambiente MATLAB i seguenti esercizi.

- Siano dati i metodi dei gradienti coniugati e di Gauss-Seidel, controllare l'implementazione di ogni passo

$$[x, \text{iter}] = \text{conj_grad}(A, b, x_0, \text{tol}, N_{\text{max}})$$

$$[x, \text{iter}] = \text{g_seidel}(A, b, x_0, \text{tol}, N_{\text{max}})$$

1. Realizzare una function MATLAB **SOR.m** per la risoluzione di sistemi lineari con il relativo metodo iterativo. Sintassi: $[x, \text{iter}] = \text{SOR}(A, b, x_0, \text{tol}, N_{\text{max}}, \text{omega})$

2. Nello script **ex2.m** utilizzare, quando possibile, i metodi iterativi di Gauss-Seidel, Gradienti coniugati e SOR implementati nei rispettivi m-file: **g_seidel.m**, **conj_grad.m** e **SOR.m**, per la risoluzione dei seguenti sistemi lineari $Ax=b$, ove

$$(a) \quad A = \begin{bmatrix} 5 & 0 & -1 & 2 \\ -2 & 4 & 1 & 0 \\ 0 & -1 & 4 & -1 \\ 2 & 0 & 0 & 3 \end{bmatrix}, \quad b = [15 \ -4 \ -8 \ 3]^T;$$

prima con precisione $\text{tol}=1e-3$ e poi $\text{tol}=1e-5$. Controllare le condizioni di convergenza dei metodi.

Soluzione esatta $x = [3 \ 1 \ -2 \ -1]^T$.

(b) A ha dimensione n, **pentadiagonale**. Le uniche diagonali non nulle di A, oltre a quella principale che ha tutti elementi pari a 4, sono la prima e la terza sopra e sotto la diagonale principale che hanno elementi pari a -1. Come sempre b è scelto in modo tale che la soluzione esatta del sistema sia $x = \text{ones}(n,1)$, $x_0 = \text{zeros}(n,1)$, $n=10,100,1000$, con precisione $\text{tol}=1e-5$ e $N_{\text{max}}=1000$, (per avere il grafico di sparsità **spy(A)**)

(c) Sistema a blocchi

$$A = \begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 0 \\ 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}$$

che ha soluzione esatta $x = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$, prima con precisione $\text{tol}=1e-3$ e poi $\text{tol}=1e-5$.

Stampare in una tabella le componenti delle tre soluzioni e il numero di iterazioni utilizzate da ciascun metodo per ottenere la soluzione a meno di una tolleranza $\text{tol}=10^{-3}$. Valutare la velocità di convergenza dei metodi nella risoluzione dei sistemi lineari dati in termini del numero di iterazioni. Calcolare inoltre la norma del vettore errore tra soluzione esatta e soluzione calcolata.

3. **Metodi diretti vs metodi iterativi.** Nello script *ex3.m* utilizzare il metodo iterativo dei Gradienti coniugati *conj_grad.m* ed il metodo diretto di eliminazione di Gauss (`\` di MATLAB) per la risoluzione dei seguenti sistemi lineari $Ax=b$, ove $A=\text{hilb}(n)$, $n=8,12,15$, e $b=A*\text{ones}(n,1)$, con precisione $\text{tol}=1e-6$, $x_0=\text{zeros}(n,1)$. Visualizzare gli errori ottenuti rispetto alla soluzione esatta e il condizionamento dei sistemi lineari.

4. **Precondizionatori.** Nello script *ex4.m* Consideriamo la matrice A di dimensione $m \times m$ ottenuta dalla discretizzazione alle differenze finite dell'operatore di Laplace sul quadrato unitario $[-1,1] \times [-1,1]$. Questa matrice è stata generata dal comando

```
G=numgrid('S',n); A=delsq(G);
```

dove $m=(n-2)^2$. A è simmetrica def. pos. e diventa mal condizionata per valori grandi di m .

- Calcolare il suo numero di condizionamento per $m=400$ e $m=1600$.
- (a) Costruire un preconditionatore M mediante fattorizzazione incompleta di Cholesky $M=R^T R$ definito da IC(0) generato dal comando

```
[R]= ichol(A, struct('type','nofill'));
```
- (b) Costruire un preconditionatore diagonale, uguale alla diagonale della matrice A :

```
M=diag(diag(A));
```
- Calcolare il numero di condizionamento di $M^{-1}A$.
- Risolvere il sistema lineare $Ax=b$ dove $b=A*\text{ones}(m,1)$ con il metodo dei gradienti coniugati con e senza i due diversi condizionatori confrontando il numero di iterazioni richieste.

```
[X,FLAG,RELRES,ITER] = pcg(A,b,1e-10,200,M)
```

RELRES: restituisce il residuo normalizzato definito come segue: $\frac{\|b - Ax\|_2}{\|b\|_2}$

- Visualizzare in un grafico il vettore `resvec` relativo a ciascuno dei metodi.

Verifica condizioni di convergenza

A partire dalla matrice A costruire:

- la matrice diagonale D avente come elementi gli elementi diagonali di A ($D=\text{diag}(\text{diag}(A))$);
- La matrice L ottenuta da A considerando solo il triangolo inferiore esclusi gli elementi diagonali, usare l'istruzione $L=\text{tril}(A,-1)$;
- La matrice U ottenuta da A considerando solo il triangolo superiore esclusi gli elementi diagonali, usare l'istruzione $U=\text{triu}(A,1)$;

Costruire le matrici della decomposizione N e P , per ciascuno dei metodi considerati Gauss-Seidel, SOR (Gauss-Seidel con rilassamento):

Gauss_Seidel $\rightarrow N = D+L, P = -U$

SOR $\rightarrow N = D+\omega L; P = ((1-\omega)*D-\omega*U);$

Costruire la matrice di Iterazione $M = \text{inv}(N) * P$; Calcolare il raggio spettrale di M , cioè l'autovalore di modulo massimo, ($\rho = \text{abs}(\text{eigs}(M, 1, 'lm'))$); oppure $\|M\|_1$