

Esercizio 1.

Per ognuno dei seguenti sistemi lineari

$$A_1 = \begin{cases} x_1 - 2x_2 + 3x_3 - x_4 = 1 \\ -x_2 + 2x_3 + x_4 = -1 \\ 2x_1 - x_2 + x_3 = 2 \\ 2x_2 + x_4 = 4 \end{cases} \quad A_2 = \begin{cases} x_1 - 0.01x_2 - 0.1x_3 - x_4 = -1.2 \\ x_1 + x_2 - 3x_3 + 2x_4 = 9 \\ -0.01x_2 + x_3 - x_4 = 3.3 \\ -x_1 + x_3 + x_4 = 5.5 \end{cases}$$

rispondere alle domande:

1. Verificare o meno la convergenza del metodo di Jacobi;
2. Verificare o meno la convergenza del metodo di Gauss-Seidel;
3. Nel caso in cui un metodo (Jacobi e/o Gauss-Seidel) risulti convergente determinare il numero di iterazioni affinché l'errore in norma infinito (componente per componente) abbia un valore minore di 10^{-4} .

Esercizio 2.

Dato il sistema lineare $\mathbf{Ax} = \mathbf{b}$, con

$$\mathbf{A} = \begin{pmatrix} 4 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & -1 & 4 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

dimostrare che i metodi di Jacobi e Gauss-Seidel convergono entrambi.

Dire inoltre, quali dei due metodi converge più rapidamente alla soluzione, fissata una tolleranza di $1.0\text{E-}8$ sulla norma del massimo, il vettore iniziale pari a $\mathbf{x}_0 = [0, 0, 0, 0, 0, 0]^T$.

Esercizio 3.

Implementare il metodo del gradiente coniugato (conjugate gradient).

Riportare l'andamento del residuo della soluzione in norma due al variare dell'iterazione corrente relativo al sistema lineare la cui matrice A_{10} è ottenuta dai comandi Matlab `A10 = full(gallery('poisson',10))`; e il termine noto b_{10} è scelto in modo tale che la soluzione esatta sia $x = [1, 1, \dots, 1]^T$.

Osservazione.

Si possono utilizzare nella risoluzione degli esercizi tutte le routine Matlab relative ad un linguaggio standard di programmazione (es. for, while, if, else, ...) ed alcune funzioni quali `nchoosek` e l'operatore di indicizzazione vettorizzata "colon" $:$.

Sono invece vietate l'utilizzo di tutte quelle routine Matlab che implementano algoritmi di calcolo numerico. Ad esempio: `roots`, `fzero`, `eig`, `inv`, `deconv`, `conv`, `pcg`, `lu`, `polyval`, `det`, routine di calcolo simbolico, `\` e `/` applicati alla risoluzione di sistemi lineari.