Metodi del Calcolo Scientifico - Progetto 1 alternativo

Francesco Trolli
889039

June 5, 2025

1 Introduzione

L'obiettivo principale di questo progetto è stato confrontare diversi metodi iterativi per la risoluzione di sistemi lineari, analizzandone le prestazioni in termini di tempo di esecuzione, numero di iterazioni e accuratezza dei risultati.

2 Descrizione del Progetto

Il progetto prevede l'implementazione e il confronto dei seguenti metodi iterativi:

- Metodo di Jacobi
- Metodo di Gauss-Seidel
- Metodo del Gradiente
- Metodo del Gradiente Coniugato

Le matrici utilizzate per i test sono state caricate da file in formato Matrix Market (`.mtx`). Per garantire la correttezza nel caricamento delle metrici è stato aggiunto il `Matrix Market banner` ad ogni file.

3 Implementazione

3.1 Caricamento delle Matrici

Le matrici sono state caricate utilizzando la funzione `mmread` del pacchetto `scipy.io`, che implementa una serie di controlli interni i quali vengono rimandati all'utente in caso di errore durante il caricamento.

3.2 Verifica delle Proprietà delle Matrici

Per ogni matrice si è verificato che fosse quadrata e che le dimensioni della matrice corrispondessero con quelle del vettore iniziale. Verificato anche per il metodo di Jacobi e Gauss-Seidel che le matrici fossero a dominanza diagonale, anche se, non essendo condizione necessaria per la convergenza, non blocca l'esecuzione del metodo.

Inoltre, per garantire la correttezza dei metodi del gradiente, è stato necessario verificare che le matrici fossero simmetriche e definite positive (SPD). Per fare ciò, è stata verificata la simmetria e la positività delle matrici prima dell'esecuzione dei metodi. In caso contrario, i metodi del gradiente non verranno eseguiti, e verrà mostrato un avviso all'utente.

3.3 Implementazione dei Metodi Iterativi

Ogni metodo iterativo è stato implementato in modo modulare, con controlli per evitare overflow e divisioni per zero. In particolare, per il metodo del gradiente, è stato aggiunto un controllo per verificare che il denominatore nel calcolo del passo non sia nullo o infinito.

4 Analisi dei Risultati del Confronto tra Metodi Iterativi

Abbiamo applicato i quattro metodi iterativi per la risoluzione di di quattro matrici sparse (`spa1`, `spa2`, `vem1`, `vem2`), tutte verificate SPD tramite decomposizione di Cholesky. Le tolleranze di arresto esaminate sono $\tau \in \{10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}\}$. I risultati, riepilogati nelle Tabelle 1 2 3 4, riportano per ciascun metodo il numero di iterazioni, il tempo di calcolo e l'errore relativo $\|x_{\text{approx}} - x_{\text{exact}}\|/\|x_{\text{exact}}\|$.

Table 1: Risultati per `spa1`.

Tolleranza	Metodo	Iterazioni	Tempo (s)	Errore relativo
10^{-4}	Jacobi	1	0.00126	3.17×10^{-16}
	Gauss–Seidel	1	0.01610	1.05×10^{-15}
	Gradiente	5465	2.09311	1.50×10^{-4}
	Gradiente Coniugato	145	0.06532	1.52×10^{-5}
10^{-6}	Jacobi	1	0.00192	3.17×10^{-16}
	Gauss–Seidel	1	0.014998	1.05×10^{-15}
	Gradiente	10143	3.62284	1.50×10^{-6}
	Gradiente Coniugato	186	0.08626	2.23×10^{-8}
10^{-8}	Jacobi	1	0.00095	3.17×10^{-16}
	Gauss–Seidel	1	0.01400	1.05×10^{-15}
	Gradiente	14829	5.35589	1.50×10^{-8}
	Gradiente Coniugato	209	0.10207	1.93×10^{-10}
10^{-10}	Jacobi	1	0.00150	3.17×10^{-16}
	Gauss–Seidel	1	0.013999	1.05×10^{-15}
	Gradiente	19517	7.61289	1.50×10^{-10}
	Gradiente Coniugato	232	0.11293	1.82×10^{-12}

Table 2: Risultati per `spa2`.

Tolleranza	Metodo	Iterazioni	Tempo (s)	Errore relativo
10^{-4}	Jacobi	1	0.01005	3.73×10^{-16}
	Gauss–Seidel	1	0.14008	5.28×10^{-16}
	Gradiente	3985	32.44910	3.39×10^{-5}
	Gradiente Coniugato	179	1.40061	2.69×10^{-6}
10^{-6}	Jacobi	1	0.00906	3.73×10^{-16}
	Gauss–Seidel	1	0.14529	5.28×10^{-16}
	Gradiente	7169	41.64086	3.43×10^{-7}
	Gradiente Coniugato	225	1.72777	1.67×10^{-8}
10^{-8}	Jacobi	1	0.00791	3.73×10^{-16}
	Gauss–Seidel	1	0.15045	5.28×10^{-16}
	Gradiente	10385	60.07608	3.45×10^{-9}
	Gradiente Coniugato	275	2.11620	1.93×10^{-10}
10^{-10}	Jacobi	1	0.00798	3.73×10^{-16}
	Gauss–Seidel	1	0.14299	5.28×10^{-16}
	Gradiente	13611	78.66256	3.45×10^{-11}
	Gradiente Coniugato	315	2.41952	2.48×10^{-12}

Table 3: Risultati per `vem1`.

Tolleranza	Metodo	Iterazioni	Tempo (s)	Errore relativo
10^{-4}	Jacobi	1	0.00230	1.23×10^{-16}
	Gauss–Seidel	1	0.04402	9.63×10^{-17}
	Gradiente	760	1.11915	6.20×10^{-3}
	Gradiente Coniugato	37	0.06996	7.24×10^{-5}
10^{-6}	Jacobi	1	0.00200	1.23×10^{-16}
	Gauss–Seidel	1	0.04310	9.63×10^{-17}
	Gradiente	1482	2.16194	6.21×10^{-5}
	Gradiente Coniugato	44	0.08540	7.61×10^{-7}
10^{-8}	Jacobi	1	0.00191	1.23×10^{-16}
	Gauss–Seidel	1	0.04308	9.63×10^{-17}
	Gradiente	2206	3.28358	6.17×10^{-7}
	Gradiente Coniugato	52	0.10033	7.00×10^{-9}
10^{-10}	Jacobi	1	0.00191	1.23×10^{-16}
	Gauss–Seidel	1	0.04509	9.63×10^{-17}
	Gradiente	2928	4.24878	6.21×10^{-9}
	Gradiente Coniugato	58	0.11170	5.21×10^{-11}

Table 4: Risultati per `vem2`.

Tolleranza	Metodo	Iterazioni	Tempo (s)	Errore relativo
10^{-4}	Jacobi	1	0.006996	1.25×10^{-16}
	Gauss–Seidel	1	0.104230	1.50×10^{-16}
	Gradiente	1082	4.72578	9.59×10^{-3}
	Gradiente Coniugato	45	0.26483	1.60×10^{-4}
10^{-6}	Jacobi	1	0.00591	1.25×10^{-16}
	Gauss–Seidel	1	0.10387	1.50×10^{-16}
	Gradiente	2208	9.64341	9.69×10^{-5}
	Gradiente Coniugato	55	0.32516	7.80×10^{-7}
10^{-8}	Jacobi	1	0.00598	1.25×10^{-16}
	Gauss–Seidel	1	0.10534	1.50×10^{-16}
	Gradiente	3336	14.54260	9.73×10^{-7}
	Gradiente Coniugato	63	0.37090	2.55×10^{-8}
10^{-10}	Jacobi	1	0.00593	1.25×10^{-16}
	Gauss–Seidel	1	0.10200	1.50×10^{-16}
	Gradiente	4466	19.82786	9.70×10^{-9}
	Gradiente Coniugato	73	0.42792	4.37×10^{-11}

4.1 Considerazioni generali

1. **Verifica SPD**: tutte le matrici (`spa1`, `spa2`, `vem1`, `vem2`) risultano effettivamente SPD, poiché superano la decomposizione di Cholesky senza errori. Questo garantisce che i metodi basati sul gradiente (gradiente semplice e gradiente coniugato) teoricamente dovrebbero convergere.

2. **Dominanza diagonale**: in tutti i test è emesso un avviso di *non dominanza diagonale* per Jacobi e Gauss–Seidel. Tuttavia, pur senza dominanza diagonale stretta, entrambi i metodi convergono *in una sola iterazione* per tutte le tolleranze e per tutte le matrici.

   - Ciò indica che la scelta di $x_0 = \mathbf{0}$ e la costruzione di $b = A\,\mathbf{1}$ rendono l'iterata di Jacobi / Gauss–Seidel immediatamente uguale alla soluzione esatta numerica. Questa è una conseguenza diretta della scelta di sistema e del vettore di partenza.
   - Se ciò non fosse desiderato per test più generali, si potrebbe variare x_0 (ad esempio un vettore casuale) per studiare la vera velocità di convergenza.

Da un punto di vista computazionale, entrambi i metodi hanno un costo per iterazione dell'ordine di $O(N^2)$, essendo basati su prodotti matrice-vettore. Tuttavia, Gauss-Seidel può essere leggermente più oneroso, perché comporta la risoluzione di un sistema triangolare inferiore ad ogni iterazione, anche se in pratica questo viene effettuato tramite una semplice sostituzione in avanti.

3. **Metodo del Gradiente**:

   - Converge in un numero di iterazioni che cresce all'abbassarsi della tolleranza (da $\sim 5 \times 10^3$ fino a $\sim 2 \times 10^4$ per `spa1`).
   - I tempi di esecuzione diventano molto lunghi su matrici `spa2` (fino a $\sim$ 78 s per $\tau = 10^{-10}$).
   - L'errore relativo diminuisce con τ (ad esempio su `spa1`: da 1.50×10^{-4} a 1.50×10^{-10}).
   - Anche su `vem1` e `vem2`, benché SPD, il gradiente semplice richiede alcune migliaia di iterazioni, con un errore finale fra 10^{-3} e 10^{-9}. Ciò conferma l'elevato numero di iterazioni tipico di un metodo del gradiente puro quando il condizionamento della matrice non è ottimale.

4. **Metodo del Gradiente Coniugato**:

   - Converge in un numero molto inferiore di iterazioni (da 37 fino a 315 circa): circa 0.1–0.4 s su matrici `vem1`, `vem2`; $\sim$ 0.07–2.4 s su `spa2`.
   - L'errore relativo scende da 7×10^{-5} fino a 10^{-12} in base alla tolleranza impostata.
   - Conferma come descritto nella teoria che su matrici SPD converge in $\leq n$ passi, e nella pratica anche molti meno.
   - Anche su sistemi non strettamente diagonali dominanti, il gradiente coniugato si dimostra estremamente efficiente: il numero di iterazioni cresce lentamente man mano che τ diminuisce ($\sim$ 4% di incremento ogni ordine di grandezza di tolleranza su `spa1`, `spa2`).

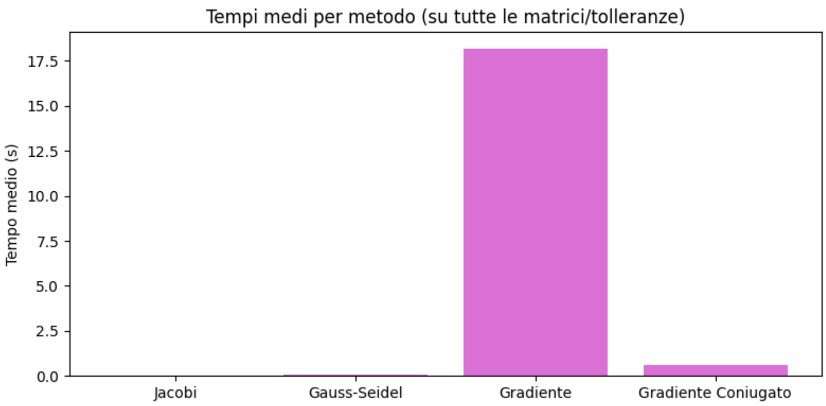

Figure 1: Tempo medio di esecuzione dei metodi (su tutte le matrici e tolleranze).

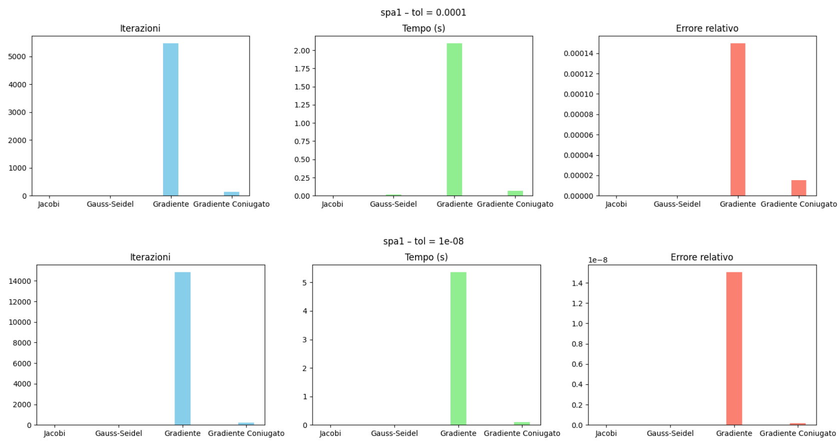

Figure 2: Confronto tra metodi su `spa1` per due tolleranze: sopra $\tau = 10^{-4}$, sotto $\tau = 10^{-8}$.

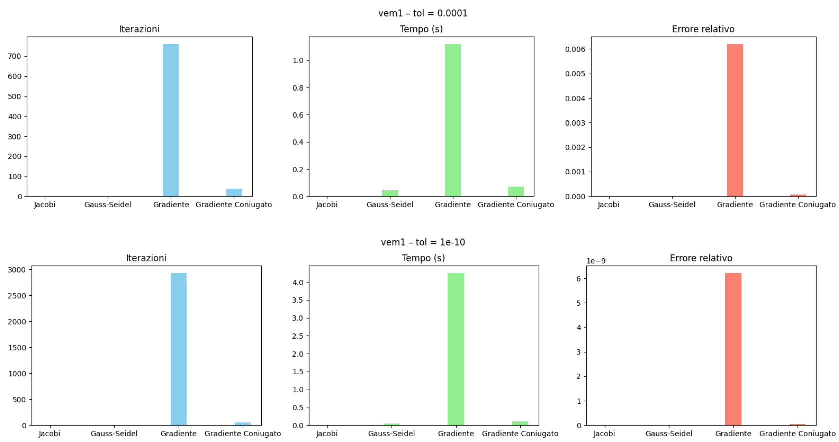

Figure 3: Confronto tra metodi su `vem1` per due tolleranze: sopra $\tau = 10^{-4}$, sotto $\tau = 10^{-10}$.

I grafici in Figura 2 e 3 mostrano la distribuzione delle iterazioni, dei tempi e degli errori per `spa1` e `vem1` a due tolleranze diverse. In tutti i casi, il gradiente coniugato appare molto più efficiente in termini di iterazioni rispetto al gradiente semplice.

5 Considerazioni finali

- **Jacobi e Gauss–Seidel**: Jacobi e Gauss-Seidel sono metodi semplici da implementare e utili per matrici di dimensioni moderate e ben condizionate. Gauss-Seidel è generalmente più veloce, ma anche più difficile da parallelizzare rispetto a Jacobi. Entrambi diventano meno competitivi quando la dimensione del sistema cresce o la matrice è mal condizionata, casi in cui metodi come il Gradiente Coniugato risultano più efficienti.

- **Gradiente**: Il metodo del gradiente è un algoritmo iterativo che, a differenza dei metodi stazionari, sfrutta il residuo per avanzare in direzione di "discesa massima" della funzione quadratica associata al sistema $Ax = b$. Inoltre sappiamo che se A è simmetrica e definita positiva (SPD), il gradiente converge, qualunque sia la scelta del vettore iniziale. Tuttavia, per avere una convergenza rapida (cioè per richiedere un numero contenuto di iterazioni), è fondamentale che il sistema abbia un buon condizionamento: in presenza di autovalori troppo distanti, il numero di iterazioni e il tempo diventano proibitivi. Nella matrice `vem2` questo fenomeno si osserva chiaramente: il gradiente richiede migliaia di iterazioni e decine di secondi per raggiungere tolleranze moderate.

- **Gradiente Coniugato**: questo metodo rappresenta il miglior compromesso per matrici SPD: convergenza rapida, pochi passi ed errore molto basso. Questo rappresenta un buon compromesso tra velocità e accuratezza. Infatti a differenza del metodo del gradiente una volta trovato un vettore ottimale $x^{(k)}$ rispetto a una direzione d non lo si deve più modificare lungo quella direzione, riducendo di gran lunga il numero di iterazioni.