Corso di Calcolo Numerico e Programmazione

Ingegneria per l'Ambiente e il Territorio

Esercitazione 3: Sulla Soluzione di Sistemi Lineari

Si vuole risolvere il sistema lineare: $Ax = b$ dove la matrice A è di dimensioni $n \times n$ e il vettore b è di dimensioni n, con $n = 900$. Gli elementi della matrice A e del vettore b sono contenuti del file `mat900.dat` e `vett900.dat` e si trovano al seguente link:

`http://www.math.unipd.it/~putti/teaching/calcolo_ambientale/index.html`

La matrice in questione è biciclica e coerentemente ordinata.

Costruire due programmi *FORTRAN* che risolvano il sistema assegnato rispettivamente con gli schemi di Jacobi e di sovrarilassamento (SOR). Si fissi la tolleranza per il criterio di uscita dal ciclo come segue:

$$\|x_{k+1} - x_k\| < \varepsilon$$

con $\varepsilon = 10^{-8}$, essendo $\|.\|$ la norma euclidea. Si utilizzi doppia precisione (`real*8`) in tutti i calcoli reali.

1. Si risolva il sistema dapprima con il metodo di Seidel ($\omega = 1$) e poi con il metodo di Jacobi.

2. Per entrambi i metodi, si stimi la costante asintotica di convergenza $\lambda_{1,(J,S)}$ utilizzando il rapporto tra la norma euclidea degli scarti $\|d_{k+1}\| = \|x_{k+1} - x_k\|$ a due iterazioni successive. Si verifichino le relazioni esistenti tra i raggi spettrali delle matrici di iterazione di Jacobi e Seidel ($\lambda_{1,J} = \lambda_{1,S}^2$).

3. Nel caso di Seidel, si riporti in un grafico semilogaritmico la norma euclidea $\|d_k\|$ in funzione di k e si calcoli $\lambda_{1,S}$ utilizzando il valore della pendenza del segmento rettilineo del grafico. Si calcoli il valore di ω_{opt} dato da:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \lambda_{1,S}}}.$$

4. Partendo da $\omega = 1.5$ e finendo con $\omega = 1.9$ con passo $\Delta\omega = 0.01$ si risolva il sistema per i diversi valori di ω[1]. Si riporti in un grafico il numero di iterazioni al variare di ω e si stimi il valore approssimato di ω_{opt}.

Si scriva una breve relazione riportando il problema risolto, il metodo utilizzato, i risultati ottenuti (compresi i grafici) e il testo del programma FORTRAN.

[1] Il modo migliore di procedere è quello di costruire una SUBROUTINE che implementa il metodo SOR per un dato valore di ω. Una volta che questo è stato verificato, si procede più facilmente a implementare un ciclo WHILE per i diversi valori di ω richiesti.

Note sull'implementazione del metodo SOR

Per implementare il metodo SOR utilizzando un qualsiasi linguaggio di programmazione (ad es. FORTRAN), si utilizza la formula del metodo SOR scritta componente per componente. La ricaviamo nel seguente modo.

Sia da risolvere il sistema

$$Ax = b \tag{1}$$

con A matrice $n \times n$, x vettore $n \times 1$ e b vettore $n \times 1$. Decomponendo la matrice in $A = L + D + U$, dove

$$l_{ij} = \begin{cases} a_{ij}, & \text{se } i < j, \\ 0, & \text{se } i \geq j. \end{cases} ; \qquad d_{ij} = \begin{cases} a_{ij}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} ; \qquad u_{ij} = \begin{cases} a_{ij}, & \text{se } i > j, \\ 0, & \text{se } i \leq j. \end{cases} \tag{2}$$

e sostituendo nel sistema iniziale (1), dopo facili calcoli si ottiene:

$$(\omega L + D)x_{k+1} = [(1-\omega)D - \omega U]x_k + \omega b \tag{3}$$

che è l'espressione dello schema SOR. Si ricorda che per $\omega = 1$ si ottiene lo schema di Seidel.

La i-esima componente dell'equazione (3) si scrive:

$$\omega \sum_{j=1}^{n} l_{ij} x_{k+1,j} + \sum_{j=1}^{n} d_{ij} x_{k+1,j} = (1-\omega) \sum_{j=1}^{n} d_{ij} x_{k,j} - \sum_{j=1}^{n} u_{ij} x_{k,j} + \omega b_i$$

Dalla (2) si osserva che gli elementi diagonali e della triangolare superiore (inferiore) della L (U) sono nulli, mentre la D ha solo gli elementi diagonali diversi da zero. La (3) si può dunque scrivere:

$$\sum_{j=1}^{n} d_{ij} x_{k+1,j} = \sum_{j=1}^{n} d_{ij} x_{k,j} + \omega \left(b_i - \sum_{j=1}^{i-1} l_{ij} x_{k+1,j} - \sum_{j=i}^{i} d_{ij} x_{k,j} - \sum_{j=i+1}^{n} u_{ij} x_{k,j} \right)$$

per cui, notando che $\sum_{j=1}^{n} d_{ij} x_j = a_{ii} x_i$ e la (2), si ottiene:

$$x_{k+1,i} = x_{k,i} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_{k+1,j} - \sum_{j=i}^{n} a_{ij} x_{k,j} \right)$$

che è la formula che si utilizza nell'implementazione numerica del metodo SOR.

Un algoritmo per l'implementazione del metodo si può riassumere nel seguente modo:

ALGORITMO SOR:

dati di input: x_k (soluzione iniziale), $IMAX$ (max. n. di iterazioni) $TOLL$ (tolleranza di uscita), scarto$= 2 * TOLL$;

DO WHILE scarto $> TOLL$ and $iter \leq IMAX$

1. $iter \leftarrow iter + 1$
2. DO $i = 1, n$
   a. som1 $= \sum_{j=1}^{i-1} a_{ij} x_{k+1,j}$
   b. som2 $= \sum_{j=i}^{n} a_{ij} x_{k,j}$
   c. $x_{k+1,i} = x_{k,i} + \frac{\omega}{a_{ii}}(b_i - \text{som1} - \text{som2})$
3. END DO
4. scarto$= ||x_{k+1} - x_k||$
5. $x_k \leftarrow x_{k+1}$

END DO