Università degli Studi di Padova

Laboratorio di Calcolo Numerico
Laboratorio 10: Esercizi su matrici e soluzione di sistemi lineari

Andrea Franceschini

E-mail: franceschini@dmsa.unipd.it

Dispense:
http://www.math.unipd.it/∼putti/teaching/calcolo_ambientale/index.html

19 Maggio 2016

Implementazione sommatoria in Fortran

Lo scopo è implementare la seguente sommatoria in Fortran (prodotto matrice vettore, componente c_i):

$$c_i = \sum_{j=1}^{n} a_{ij} b_j \tag{1}$$

In Fortran, la (1) diventa:

```
do i = 1,m
    som1 = 0.d0
    do j = 1,n
        som1 = som1 + A(i,j) * b(j)
    end do
    c(i) = som1
end do
```

Esercizio

1. Implementare la norma euclidea di un vettore, $\|x\|_2 = \sqrt{\sum_{i=1}^{n} (x_i)^2}$, tramite una function (chiamata "norm_eucl").
2. (ESERCIZIO PER CASA) Scrivere una function che calcoli la norma di Frobenius di A,

$$\|A\|_F = \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{n} (a_{ij})^2}$$

3. (ESERCIZIO PER CASA) Scrivere una subroutine che calcoli la matrice $G = AA^T$. In componenti: $G_{ij} = \sum_{k=1}^{n} a_{ik} a_{jk}$.

Metodi iterativi stazionari per la risoluzione di sistemi lineari

Dati A matrice quadrata $n \times n$ non singolare e b vettore ($b \in \mathbb{R}^n$), si vuole trovare un'approssimazione del vettore $x \in \mathbb{R}^n$ tale che

$$Ax = b$$

Esempio

$$A = \begin{pmatrix} 5.0 & 1.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 3.0 & 0.0 & 1.0 & -1.0 \\ 1.0 & 0.0 & 5.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 3.0 & 1.0 \\ 0.0 & -1.0 & 0.0 & 1.0 & 3.0 \end{pmatrix} \qquad b = \begin{pmatrix} 7.0 \\ 4.0 \\ 6.0 \\ 5.0 \\ 3.0 \end{pmatrix}$$

Calcolare x con gli algoritmi di Jacobi e Gauss-Seidel e verificare che $Ax = b$.

Implementazione dell'algoritmo di Jacobi

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

Condizione di terminazione: $\|x^{(k+1)} - x^{(k)}\| < toll$ oppure $k > itmax$.

do while (scarto **.gt.** toll **.and.** k **.lt.** itmax)
    k $\leftarrow$ k + 1
    do i = 1,n
        som1 = $\sum_{j=1}^{i-1} a_{ij} x_j^{(k)}$
        som2 = $\sum_{j=i+1}^{n} a_{ij} x_j^{(k)}$
        $x_i^{(k+1)} = \frac{1}{a_{ii}} (b_i -$ som1 $-$ som2$)$
    end do
    scarto = $\left\| x^{(k+1)} - x^{(k)} \right\|_2$
    $x^{(k)} \leftarrow x^{(k+1)}$
end do

Implementazione dell'algoritmo di Seidel

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

do while (scarto **.gt.** toll **.and.** k **.lt.** itmax)

  k $\leftarrow$ k + 1

  do i = 1,n

    som1 = $\sum_{j=1}^{i-1} a_{ij} \boxed{x_j^{(k+1)}}$

    som2 = $\sum_{j=i+1}^{n} a_{ij} x_j^{(k)}$

    $x_i^{(k+1)} = \frac{1}{a_{ii}} (b_i -$ som1 $-$ som2$)$

  end do

  scarto = $\left\| x^{(k+1)} - x^{(k)} \right\|_2$

  $x^{(k)} \leftarrow x^{(k+1)}$

end do

Matrice di prova

$$A = \begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \quad b = \begin{pmatrix} 100 \\ 100 \\ 200 \\ 0 \\ 0 \\ 100 \\ 0 \\ 0 \\ 100 \end{pmatrix}$$