

Richiami sul metodo di Jacobi

Dobbiamo risolvere il sistema $A\mathbf{x} = \mathbf{b}$.

Lo schema di Jacobi è:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + D^{-1}\mathbf{r}_k$$

D = la matrice diagonale contenente gli elementi diagonali di A

\mathbf{r}_k = il residuo al passo $k = \mathbf{b} - A\mathbf{x}_k$.

Introducendo la matrice di iterazione di Jacobi:

$$E_J = I - D^{-1}A = -D^{-1}(L + U)$$

(L e U sono le matrici triangolare bassa e triangolare alta di A , con elementi della diagonale principale nulli, tali che $L + D + U = A$) lo schema di Jacobi si può anche scrivere come

$$\mathbf{x}_{k+1} = -D^{-1}(L + U)\mathbf{x}_k + D^{-1}\mathbf{b}$$

o, equivalentemente,

$$\mathbf{x}_{k+1} = D^{-1}[-(L + U)\mathbf{x}_k + \mathbf{b}]$$

In forma estesa, si ha:

per $i = 1, n$

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

\uparrow \uparrow

o, equivalentemente, per $i = 1, n$

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

\uparrow \uparrow \uparrow

In FORTRAN

Se vogliamo tradurre quest'ultimo algoritmo in un programma FORTRAN, dobbiamo calcolare due sommatorie (quelle relative a Lx_k e a Ux_k) prima di aggiornare la componente i -sima del vettore che approssima la soluzione del sistema lineare.

Sia n la dimensione del sistema da risolvere, A la matrice, b il vettore termine noto.

Dati di input:

- x_k (soluzione iniziale),
- IMAX (max. n. di iterazioni),
- TOLL (tolleranza di uscita),
- scarto = $2 * \text{TOLL}$.

Fino a che (scarto $>$ TOLL) e (iter \leq IMAX)

1. iter = iter + 1

2. **Per** $i = 1, n$

a. som1 =
$$\sum_{j=1}^{i-1} a_{ij} x_{k,j}$$

b. som2 =
$$\sum_{j=i+1}^n a_{ij} x_{k,j}$$

c. $x_{k+1,i} = \frac{1}{a_{ii}} (b_i - \text{som1} - \text{som2})$

3. **Fine Per**

4. scarto = $\|x_{k+1} - x_k\|_2$

5. $x_k = x_{k+1}$

Esempio: calcolo della sommatoria al punto (2.a)

```
.....      do i=1,n
.....          som1=0.d0
.....          .....
.....          do j=1,i-1
.....              som1=som1+A(i,j)*xold(j)
.....          end do
.....          .....
.....      end do
```

Stima della velocità asintotica di convergenza

Utilizzando la norma euclidea dello scarto al passo k e al passo $k + 1$, possiamo calcolare una stima del fattore di convergenza del metodo (vale a dire il modulo dell'autovalore dominante della matrice di iterazione di Jacobi).

Dalla teoria, sappiamo che, per $k \rightarrow \infty$, l'errore

$$\mathbf{e}_{k+1} = E\mathbf{e}_k \approx \lambda_1^{k+1} c_1 \mathbf{v}_1 \Rightarrow \lim_{k \rightarrow \infty} \frac{\|\mathbf{e}_{k+1}\|}{\|\mathbf{e}_k\|} = |\lambda_1|.$$

Per $k \rightarrow \infty$ lo scarto rappresenta una buona approssimazione per l'errore: introducendo lo scarto $sc(i) = x_{new}(i) - x_{old}(i)$, per $i=1, n$ e calcolando la norma euclidea dello scarto a due iterazioni successive, possiamo stimare il fattore di convergenza $M = |\lambda_1|$ come rapporto di queste due norme.

Analogamente (considerando la definizione di velocità asintotica di convergenza $R = -\log_{10} (|\lambda_1|)$), possiamo stimare R applicando $-\log_{10}$ a tale rapporto.

In FORTRAN, dando alle variabili `scarto` e `scarto_old` il significato della norma euclidea dello scarto al passo $k + 1$ (nuovo) e al passo k (old), possiamo scrivere, dopo aver valutato `xnew` :

```
do i=1,n
    sc(i)=xnew(i) - xold(i)
end do
scarto=euc(n,sc)  %euc: function su norma euclidea
asint=scarto/scarto_old
R=-log10(asint)
```

Stampa dei risultati parziali

Durante ciascuna iterazione, vengono calcolati `xnew`, `scarto`, `asint`, `R`. Ad ogni iterazione ci facciamo stampare `scarto`, `asint`, `R`.

A tal fine possiamo pensare di stampare i valori secondo un certo **formato**, per esempio con un formato numerico con 8 cifre decimali dopo la virgola, o con un formato scientifico con 8 cifre nella mantissa...

Vediamo un esempio:

```
.....write(16,100) iter, scarto, asint, R  
.100..format(i6, e20.8, 2f14.8)
```

Cenni sui vari tipi di formato

iN	stampa di numeri interi usando N caratteri
eM.N	stampa di numeri reali in formato esponenziale usando M caratteri, di cui N per la mantissa
fM.N	stampa di numeri reali in formato fisso M caratteri sono in virgola fissa N caratteri sono per le cifre decimali
aN	formato alfanumerico per stringhe di N caratteri
Nx	stampa N caratteri bianchi (vuoti)
/	serve per andare a capo nella stampa

Esempi

```
write(16,99) 'iterazione', iter
99  format(1x,a10,3x,i4)

write(16,100) iter,scarto,asint,r
100 format(i6,e20.8,2f14.8)

write(16,101) iter, scarto, asint, R
101 format(1x, i4, 2x, e16.8, 2x, 2f14.8)
```

in output ↓↓

```
iterazione          7
 21      0.53629231E-02      0.80901699      0.09204236
 21      0.53629231E-02      0.80901699      0.09204236
```