

LABORATORIO DI CALCOLO NUMERICO

Laurea in Statistica e Informatica

Esercitazione di algebra lineare numerica: I

Prof. Stefano De Marchi

Padova, November 3, 2010

1 Operazioni vettoriali in Matlab

In questo paragrafo presentiamo alcuni comandi, sotto forma di esempi, utili in Matlab per lavorare con matrici.

1.1 Operazioni su singole righe o colonne

- Data una matrice A di ordine n , le istruzioni

```
for i = 1:n  
    A(i,j) = A(i,j)+1;  
end
```

possono essere sostituite da

```
A(1:n,j) = A(1:n,j)+1;
```

oppure, usando l'operatore :

```
A(:,j) = A(:,j)+1;
```

- È possibile *scambiare righe o colonne*. Per esempio, l'istruzione

```
A = B([1 3 2],:);
```

crea una matrice A che ha per prima riga la prima riga di B , per seconda la terza di B e per terza la seconda di B . Analogamente, l'istruzione

```
A = B(:,[1:3 5:6]);
```

crea una matrice A che ha per colonne le prime tre colonne di B e poi la quinta e la sesta di B .

- È possibile *concatenare matrici*. Per esempio, l'istruzione

```
U = [A b];
```

crea una matrice U formata da A e da un'ulteriore colonna b (ovviamente le dimensioni di A e b devono essere compatibili).

- È possibile *assegnare lo stesso valore ad una sottomatrice*. Per esempio, l'istruzione

```
A(1:3,5:7) = 0;
```

pone a zero la sottomatrice formata dalle righe dalla prima alla terza e le colonne dalla quinta alla settima.

- Un altro comando utile per la manipolazione di matrici è `max`. Infatti, nella forma

```
[M, i] = max(A(2:7,j));
```

restituisce l'elemento massimo M nella colonna j -esima di A (dalla seconda alla settima riga) e la posizione di tale elemento nel vettore $[a_{2,j}, a_{3,j}, \dots, a_{7,j}]^T$.

Conclusione. Tutte le istruzioni vettoriali che sostituiscono cicli `for` sono da preferirsi dal punto di vista dell'efficienza computazionale in Matlab!

2 Sostituzioni all'indietro

L'algoritmo delle sostituzioni all'indietro per la soluzione di un sistema lineare $Ux = b$, con U matrice triangolare superiore, può essere scritto

```
function x = BS(U,b)
% x = BS(U,b)
n = length(b);
x = b;
x(n) = x(n)/U(n,n);
for i = n-1:-1:1
    for j = i+1:n
        x(i) = x(i)-U(i,j)*x(j);
    end
    x(i) = x(i)/U(i,i);
end
```

Table 1: Sostituzione all'indietro

Per quanto visto nel paragrafo precedente, le istruzioni

```
for j = i+1:n
    x(i) = x(i)-U(i,j)*x(j);
end
x(i) = x(i)/U(i,i);
```

possono essere sostituite da

```
x(i) = (x(i)-U(i,i+1:n)*x(i+1:n))/U(i,i);
```

ove, in questo caso, l'operatore $*$ è il prodotto scalare tra vettori.

Generalizzando al caso di più termini noti b_1, b_2, \dots, b_m , si ottiene

```
function X = BSS(U,B)
%
% X = BSS(U,B)
%
n = length(B);
X = B;
X(:, :) = X(:, :)./U(:, n);
for i = n-1:-1:1
    X(:, :)=(X(:, :) - U(:, i+1:n)*X(:, i+1:n))/U(:, i);
end
```

Table 2: Sostituzione all'indietro per sistemi

Questa function risolve

$$UX = B$$

cioè

$$UX_i = b_i, \quad i = 1, 2, \dots, m.$$

Esercizi proposti

- Si consideri la matrice di Hilbert di ordine n (in Matlab, `H=hilb(n)`, con n scelto a piacere) e il sistema lineare

$$Hx = b.$$

Si consideri come soluzione $x = (1, \dots, 1)^T$.

- Risolvere $Hx = b$ con un metodo diretto (MEG e/o Cholesky).
- Si perturbi quindi b per mezzo del vettore $\delta b = (0, \dots, 0, 1.e - 4)^T$. Risolvere

$$H\hat{x} = b + \delta b.$$

Stimare il numero di condizionamento in norma 2 di H , $\kappa(H)$ mediante la relazione

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \kappa(H) \frac{\|\delta b\|}{\|b\|}.$$

2. Calcolare la fattorizzazione di Cholesky MM^T di

$$A = \begin{bmatrix} 8 & 1 & 2 & 0.5 & 2 \\ 1 & 0.5 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 0 \\ 0.5 & 0 & 0 & 7 & 0 \\ 2 & 0 & 0 & 0 & 16 \end{bmatrix}.$$

3. Si risolvano, mediante il metodo di eliminazione gaussiana e l'algoritmo delle sostituzioni all'indietro, i sistemi lineari

$$A_i x_i = b_i, \quad A_i = (A_1)^i, \quad i = 1, 2, 3, 4$$

con

$$A_1 = \begin{pmatrix} 15 & 6 & 8 & 11 \\ 6 & 6 & 5 & 3 \\ 8 & 5 & 7 & 6 \\ 11 & 3 & 6 & 9 \end{pmatrix}$$

e b_i scelto in modo che la soluzione esatta sia $x_i = [1, 1, 1, 1]^T$. Per ogni sistema lineare si calcoli l'errore in norma-2 e il numero di condizionamento della matrice (con il comando `cond`). Si produca infine un grafico logaritmico-logaritmico che metta in evidenza la dipendenza lineare dell'errore dal numero di condizionamento.