Laboratorio di Analisi Numerica
Lezione 5

Federico Poloni <f.poloni@sns.it>

15 Dicembre 2010

Quantità di esercizi: in questa dispensa ci sono *più esercizi* di quanti uno studente medio riesce a farne durante una lezione di laboratorio, specialmente tenendo conto anche degli esercizi facoltativi. Questo è perché sono pensate per "tenere impegnati" per tutta la lezione anche quegli studenti che già hanno un solido background di programmazione.
Quindi fate gli esercizi che riuscite, partendo da quelli *non* segnati come facoltativi, e non preoccupatevi se non li finite tutti!

1 Fattorizzazione LU

Esercizio 1. Scrivete una `function [L,U]=my_lu(A)` che calcoli la fattorizzazione LU di una matrice A.

Hint: quali sono i calcoli da eseguire, per esempio al passo 2?

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ * & 1 & 0 & 0 & 0 \\ * & 0 & 1 & 0 & 0 \\ * & 0 & 0 & 1 & 0 \\ * & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} * & * & * & * & * \\ 0 & U_{22} & * & * & * \\ 0 & U_{32} & * & * & * \\ 0 & U_{42} & * & * & * \\ 0 & U_{52} & * & * & * \end{pmatrix}$$

Al posto degli elementi rossi di L, sostituiamo $L_{i2} = \frac{U_{i2}}{U_{22}}$. Al posto degli elementi rossi di U, sostituiamo $U_{ij} = U_{ij} - L_{i2}U_{2j}$. Nota che in questo modo gli elementi rossi sulla seconda colonna (U_{i2}) diventano 0.

```
octave:47> M=4*eye(5)+ones(5,5)
M =

  5 1 1 1 1
  1 5 1 1 1
  1 1 5 1 1
  1 1 1 5 1
  1 1 1 1 5
```

```
octave:48> [L,U]=my_lu(M)
octave:49> L*U-M
ans =

 Columns 1 through 3:

   0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
   0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
   0.00000000000000e+00 0.00000000000000e+00 -8.88178419700125e-16
   0.00000000000000e+00 0.00000000000000e+00 2.22044604925031e-16
   0.00000000000000e+00 0.00000000000000e+00 2.22044604925031e-16

 Columns 4 and 5:

   0.00000000000000e+00 0.00000000000000e+00
   0.00000000000000e+00 0.00000000000000e+00
   0.00000000000000e+00 0.00000000000000e+00
   0.00000000000000e+00 2.22044604925031e-16
   2.22044604925031e-16 0.00000000000000e+00
```

Esercizio 2. Scrivete una `function x=sys_solve(A,b)` che risolva un sistema lineare generico utilizzando la fattorizzazione LU, `inf_solve` e `sup_solve`.

Potete utilizzare le funzioni della volta scorsa per risolvere sistemi triangolari, oppure copiate e incollate le seguenti.

```
function x=inf_solve(L,b)
%risolve un sistema con L triangolare inferiore
  n=size(L)(1);
  x=b; %x "vettore di accumulatori"
  for i=1:n
    x(i)=x(i)/L(i,i);
    x(i+1:n)=x(i+1:n) - L(i+1:n,i)*x(i);
  endfor
endfunction
```

```
function x=sup_solve(U,b)
%risolve un sistema con U triangolare superiore
  n=size(U)(1);
  x=b; %x "vettore di accumulatori"
  for i=n:-1:1
    x(i)=x(i)/U(i,i);
    x(1:i-1)=x(1:i-1) - U(1:i-1,i)*x(i);
  endfor
endfunction
```

Esercizio 3 (facoltativo). Capite come funziona questa versione di `inf_solve` e `sup_solve`. Notate che i calcoli che esegue non sono identici alla versione della scorsa lezione! Ricordate che la formula è

$$x_i = \frac{b_i - \sum_{j=0}^{i-1} L_{ij} x_j}{L_{ii}}, \quad x = 1 \dots n.$$

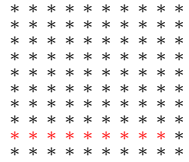
2 Esperimenti numerici

Esercizio 4. Testare i seguenti metodi di soluzione di un sistema lineare $Ax = b$:

- `sys_solve`
- Il comando di Octave `x=inv(A)*b`, che calcola la matrice inversa e la moltiplica per b.
- Il comando di Octave `x=A\b`: il comando `\` (barra rovesciata) serve proprio per risolvere sistemi lineari, ed è basato sulla fattorizzazione LU con pivoting parziale (che vedrete la volta prossima a lezione).

Per testarli, utilizzate le seguenti matrici:

- La matrice `M1=9*eye(10)+ones(10)`, che è dominante diagonale.
- La matrice `M2=rand(10)`, che è una matrice con elementi casuali — può essere abbastanza mal condizionata! Potete controllare il condizionamento con il comando `cond(M2)`.
- La matrice data da `M3=M1;M3(9,1:9)=0`, che ha una riga quasi tutta di zeri che rende la sottomatrice principale 9×9 singolare (e quindi non ammette fattorizzazione LU).
- La matrice data da `M4=M2;M4(9,1:9)=sum(M4(1:8,1:9))`:

  gli elementi in rosso sono ognuno la somma degli otto elementi che stanno direttamente sopra di esso; quindi la sottomatrice principale 9×9 è singolare (ma lavorando con i numeri floating point...).
- La matrice data da `M5=M2;M5(10,1:10)=sum(M5(1:9,1:10))`: l'ultima riga è la somma delle 9 precedenti, quindi la matrice è singolare (ma lavorando con i numeri floating point...).

- La matrice di Hilbert `hilb(10)`. Riuscite a spiegare il numero di cifre significative ottenute in base al suo numero di condizionamento, che potete calcolare con `cond(hilb(10))`?

Ponete `v=transpose(1:10)` (vettore conentente i numeri da 1 a 10 in ordine); per ognuna di queste matrici, calcolate `b`i`=v` (per $i = 1, \dots, 6$), e andate a risolvere il sistema `M`i`*x=b`i. La soluzione esatta di questo sistema è v; di quanto si discostano le soluzioni calcolate?

Esercizio 5 (facoltativo). Guardate la fattorizzazione LU di M4. `U(9,9)` è molto piccolo; perché? `U(10,10)` è molto grande; perché?

3 Facoltativo: eliminazione di Gauss "senza L"

Si possono riorganizzare i calcoli in modo che il calcolo di L venga effettuato implicitamente; questo probabilmente è equivalente a come avete visto l'eliminazione di Gauss lo scorso anno ad algebra lineare. Lavoriamo sulle equazioni anziché sulle matrici:

$$\begin{array}{llll} a_{11}x_1 & +a_{12}x_2 & +a_{13}x_3 & = b_1. \\ a_{21}x_1 & +a_{22}x_2 & +a_{23}x_3 & = b_2. \\ a_{31}x_1 & +a_{32}x_2 & +a_{33}x_3 & = b_3. \end{array}$$

Cerchiamo di eliminare x_1 dalla seconda e terza equazione:

$$\begin{array}{llll} a_{11}x_1 & +a_{12}x_2 & +a_{13}x_3 & = b_1. \\ a_{21}x_1 & +a_{22}x_2 & +a_{23}x_3 - \frac{a_{21}}{a_{11}}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3) & = b_2 - \frac{a_{21}}{a_{11}}b_1. \\ a_{31}x_1 & +a_{32}x_2 & +a_{33}x_3 - \frac{a_{31}}{a_{11}}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3) & = b_3 - \frac{a_{31}}{a_{11}}b_1. \end{array}$$

Quindi reiteriamo, fino a trasformare A in una matrice triangolare.

Prendiamo come riferimento per l'analisi il primo passo. Come vengono modificati quindi la matrice dei coefficienti (A) e il termine noto (b) che stiamo tenendo in memoria? Sulla matrice A facciamo le stesse operazioni che abbiamo fatto la scorsa lezione nel calcolo della fattorizzazione LU. Come abbiamo visto, la quantità che dobbiamo sottrarre ad A è una matrice di rango 1:

`A(2:3,1:3) = A(2:3,1:3) -` $\begin{bmatrix} \frac{a_{21}}{a_{11}} \\ \frac{a_{31}}{a_{11}} \end{bmatrix} * \begin{bmatrix} a_{11} & a_{12} & a_{13} \end{bmatrix}$

In più, dobbiamo effettuare un calcolo simile anche su b:

`b(2:3) = b(2:3) -` $\begin{bmatrix} \frac{a_{21}}{a_{11}} \\ \frac{a_{31}}{a_{11}} \end{bmatrix} * b_1$

Possiamo anche interpretare i calcoli effettuati in termini di prodotti di matrici: nel calcolo della fattorizzazione LU ottenevamo L^{-1} come prodotto di matrici parziali $L_n L_{n-1} \cdots L_1$, ognuna della forma

$$\begin{bmatrix} I & & \\ & 1 & \\ & v & I \end{bmatrix},$$

ora quello che facciamo è effettuare subito il prodotto con la matrice parziale

$$\begin{bmatrix} 1 & & \\ -\frac{a_{21}}{a_{11}} & 1 & \\ -\frac{a_{31}}{a_{11}} & 0 & 1 \end{bmatrix}$$

in modo da trasformare il sistema nel sistema equivalente (che ha la stessa soluzione)

$$L_1 A x = L_1 b,$$

Al termine degli n passi, abbiamo trasformato il sistema in un sistema equivalente ma con matrice dei coefficienti triangolare, e questo lo sappiamo risolvere.

Esercizio 6 (facoltativo). Scrivere una `function x=sys_solve2(A,b)` che risolva un sistema lineare con il metodo qui sopra.

4 Facoltativo: soluzione di un sistema lineare con fattorizzazione QR implicita (Householder)

Una trasformazione di Householder è una matrice ortogonale della forma $H_v = I - \frac{2vv^T}{\|v\|_2^2}$.

Fissato x, esiste una scelta di v che determina una H_v che "porta" x in un multiplo di e_1, ed è

$$x \pm \|x\|_2 e_1$$

In particolare, v è uguale a x tranne per il primo elemento, che vale

$$v_1 = x_1 - \|x\|_2 = \frac{-(x_2^2 + \cdots + x_n^2)}{x_1 + \|x\|_2}$$

oppure

$$v_1 = x_1 + \|x\|_2 = \frac{-(x_2^2 + \cdots + x_n^2)}{x_1 - \|x\|_2}.$$

In una di queste, il denominatore soffre di errori di cancellazione; quale? (dipende dal segno di $x_1 \dots$)

Esercizio 7 (facoltativo). Scrivete una funzione `v=householder_vector(x)` che calcoli v con il segno giusto.

Esercizio 8 (facoltativo). Scrivete una funzione `function x=qr_solve(A,b)` che risolva un sistema lineare attraverso la fattorizzazione QR: per esempio, al primo passo, invece di trovare una matrice triangolare L_1 che mandi la prima riga di A in un multiplo di e_1 (come facevate in `sys_solve2`), dovete trovare una matrice di Householder H_1 che faccia lo stesso lavoro e applicarla sia a A che a b per ottenere un sistema equivalente:

$$H_1 A x = H_1 b.$$

Esercizio 9 (facoltativo). Cosa succede se usate il segno sbagliato nella funzione `householder_vector`, o se usate sempre lo stesso segno? Testare provando a risolvere qualche sistema lineare.