Risoluzione di più sistemi con la stessa matrice

Data $A \in \mathbb{R}^{n\times n}$ e $\mathrm{b} \in \mathbb{R}^n$,

calcolare x e z : $\quad A\mathrm{x} = \mathrm{b}, \qquad A\mathrm{z} = \mathrm{c}$

costo del MEG $\quad \left(\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n\right) \quad + \quad \left(\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n\right)$

---

costo totale $= \frac{4}{3}n^3 + 3n^2 - \frac{7}{3}n$

Obiettivo: separare le operazioni che agiscono su A da quelle che agiscono su b in modo che si lavori su A una volta sola e non si ripetano calcoli inutili.
NON voglio pagare $\sim \frac{4}{3}n^3$ operazioni, bensì $\sim \frac{2}{3}n^3$ (riporto solo l'ordine massimo)

Partendo dal MEG

for $k = 1, \ldots, n-1$
  pivoting;
  for $i = k+1, \ldots, n$
    $m_{ik} = A_{ik}/A_{kk}$;
    for $j = k+1, \ldots, n$
      $A_{ij} = A_{ij} - m_{ik} A_{kj}$;
    end
    $\boxed{b_i = b_i - m_{ik} b_k}$ $\longrightarrow$ vogliamo estrarre queste operazioni dal MEG;
  end
end

Voglio separare il lavoro su A da quello su b.
L'idea consiste nel salvare i moltiplicatori m_{ik} in una matrice e posticipare il lavoro sul termine noto **b** prima di risolvere il sistema triangolare superiore.

L=matrice dei moltiplicatori

Osservo che i moltiplicatori m_{ik} sono definiti con $k = 1, \ldots, n-1$ e $i = k+1, \ldots, n$, le posizioni degli elementi nel triangolo inferiore di A. Li posso salvare al posto degli elementi nulli di A

$$\begin{bmatrix} . & . & \cdots & \cdots & . \\ m_{21} & . & \cdots & \cdots & . \\ m_{31} & m_{32} & . & \cdots & . \\ \vdots & & \ddots & \ddots & . \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & . \end{bmatrix}$$

MEG

% $(A, \mathrm{b}) \to [U, \mathrm{b}]$

for $k = 1, \ldots, n-1$

  (pivoting);

  for $i = k+1, \ldots, n$

    $m_{ik} = A_{ik}/A_{kk}$;

    for $j = k+1, \ldots, n$

      $A_{ij} = A_{ij} - m_{ik} A_{kj}$;

    end

    $b_i = b_i - m_{ik} b_k$;

  end

end

U=triu(A);

lavoro solo su A

% $A \to [L,\ U]$

for $k = 1, \ldots, n-1$

  (pivoting);

  for $i = k+1, \ldots, n$

    $A_{ik} = A_{ik}/A_{kk}$;

    for $j = k+1, \ldots, n$

      $A_{ij} = A_{ij} - A_{ik} A_{kj}$;

    end

  end

end

U= triu(A);

L= tril(A,-1)+eye(n);

Fattorizzazione LU di A

Si dimostra (con qualche conto) che le matrici L e U costruite con l'algoritmo a destra della pagina precedente sono tali che:

$$\boxed{A = L \cdot U}$$

ovvero L e U sono due fattori di A.

La scomposizione di A così ottenuta si chiama fattorizzazione LU della matrice A.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \ell_{21} & 1 & \dots & 0 \\ \vdots & & \ddots & 0 \\ \ell_{n1} & \ell_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ & & \ddots & \vdots \\ 0 & & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Risoluzione di $Ax = b$ con LU

Dopo aver calcolato le matrici L e U tali che $A = L \cdot U$, osserviamo che:

$$Ax = b \quad \Leftrightarrow L \underbrace{Ux}_{y} = b \Leftrightarrow \begin{cases} Ly = b \\ Ux = y \end{cases}$$

Se `lufact` è la function che calcola L e U, la sequenza di istruzioni per risolvere il sistema è:

```
A=[.  .  .];          % inizializzo A
b=[.  .  .];          % inizializzo b
[L,U]=lufact(A);      % fattorizzo A
y=forsub(L,b);        % risolvo Ly=b
x=backsub(U,y);       % risolvo Ux=y
```

Implementazione della fattorizzazione LU `lufact.m`

Input: $A \in \mathbb{R}^{n\times n}$ non singolare
Output: L, U
Algoritmo (senza pivotazione):

for $k = 1, \ldots, n-1$
  for $i = k+1, \ldots, n$
    $A_{ik} = A_{ik}/A_{kk}$;
    for $j = k+1, \ldots, n$
      $A_{ij} = A_{ij} - A_{ik}A_{kj}$;
    end
  end
end

U=triangolo sup di A;
L=triangolo inf di A con 1 sulla diagonale
Comandi Matlab:

```
L=tril(A,-1)+eye(n);  % estraggo tri. sotto diag princ
                      % e sommo l'identita';
U=triu(A);            % estraggo tri. sup. di A
```

Risoluzione di un sistema lineare con fattorizzazione LU

Per risolvere $Ax = b$:

		costo:
1.	calcolo L e U tali che $L\ U = A$	$\frac{2}{3}n^3 - \frac{1}{2}n^2 - \frac{7}{6}n$
2.	calcolo $\mathbf{y}$ risolvendo $Ly = b$	n^2
3.	calcolo $\mathbf{x}$ risolvendo $Ux = y$	n^2

$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n$$

Il costo totale per risolvere un solo sistema $Ax = b$ con la fattorizzazione LU è identico al costo del MEG.

Nota: Il vantaggio si osserva se devo risolvere più di un sistema lineare con la stessa matrice perché la fattorizzazione LU viene eseguita una sola volta.

Se devo risolvere due sistemi lineari

$$Ax = b \qquad e \qquad Az = c$$

utilizzando la fattorizzazione LU, il costo è:

		costo:
1.	una fattorizzazione $L\ U = A$	$\frac{2}{3}n^3 - \frac{1}{2}n^2 - \frac{7}{6}n$
2.	2 sist. tri. per il primo sistema	$2n^2$
3.	2 sist. tri. per il secondo sistema	$2n^2$

$$\frac{2}{3}n^3 + \frac{7}{2}n^2 - \frac{7}{6}n$$

Ricordo che per risolvere due sistemi lineari con MEG servivano $\frac{4}{3}n^3 + 3n^2 - \frac{7}{3}n$ operazioni.

Esercizio su fattorizzazione LU

(eslu)
Si vuole risolvere il sistema lineare $Ax = b$ con

$$A = \begin{pmatrix} 10 & 4 & 3 & -2 \\ 2 & 20 & 20 & -1 \\ 3 & -6 & 4 & 3 \\ -3 & 0 & 3 & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 5 \\ 24 \\ 13 \\ -2 \end{pmatrix}$$

con la fattorizzazione LU.
Verificare la correttezza della soluzione, richiamando MEG (o \ di matlab).

Fattorizzazione LU con pivotazione

Dobbiamo salvare gli scambi effettuati su A, per poterli applicare anche al termine noto in un secondo momento.
Utilizziamo un'altra matrice P, inizialmente uguale all'identità.
Input: $A \in \mathbb{R}^{n\times n}$ non singolare
Output: L, U, P
Algoritmo (con pivotazione):

for $k = 1, \ldots, n-1$
  trovare r t.c. $|A_{rk}| = \max_{k\le i\le n} |A_{ik}|$;
  scambiare riga r di A con riga k di A;
  scambiare riga r di P con riga k di P;
  for $i = k+1, \ldots, n$
    $A_{ik} = A_{ik}/A_{kk}$;
    for $j = k+1, \ldots, n$
      $A_{ij} = A_{ij} - A_{ik}A_{kj}$;
    end
  end
end
U=triangolo sup di A;
L=triangolo inf di A con 1 sulla diagonale

Matrice P della pivotazione

P è tale che:

$$L\ U = P\ A$$

Essendo una permutazione della matrice identità, è sicuramente non singolare.
Quindi:

$$A\mathrm{x} = \mathrm{b} \quad \Leftrightarrow P A\mathrm{x} = P\mathrm{b} \Leftrightarrow LU\mathrm{x} = P\mathrm{b} \Leftrightarrow \begin{cases} L\mathrm{y} = P\mathrm{b} \\ U\mathrm{x} = \mathrm{y} \end{cases}$$

`lufact` è la nostra function LU, con `piv=0` se non vogliamo la pivotazione, `piv=1` se vogliamo la pivotazione.

```
A=....;
b=....;
[L,U,P]=lufact(A,piv);
y=forsub(L,P*b);    % risolvo Ly=Pb
x=backsub(U,y);   % risolvo Ux=y
```

Function lu di matlab

La function `lu` di matlab implementa la fattorizzazione LU sempre con pivotazione.
Per risolvere un sistema lineare $Ax = b$ con le function di matlab:

```
A=....;
b=....;
[L,U,P]=lu(A);
y=L\(P*b);    % risolvo Ly=Pb
x=U\y;   % risolvo Ux=y
```