Esercitazione 1
Soluzione di Sistemi di Equazioni Lineari: Metodi Diretti

Il metodo di fattorizzazione LU (con pivoting)

Data una matrice quadrata A di dimensione $n \times n$ non singolare è possibile fattorizzarla con il prodotto di due matrici L ed U, dove L è una matrice triangolare inferiore ed U è una matrice triangolare superiore. Tale fattorizzazione permette di risolvere un sistema lineare del tipo

$$A\mathbf{x} = \mathbf{b}, \text{ con } A = LU.$$

Una volta calcolata la fattorizzazione LU di A si può risolvere il sistema $A\mathbf{x} = \mathbf{b}$ risolvendo in sequenza i due sistemi triangolari

$$L\mathbf{y} = \mathbf{b}, \quad U\mathbf{x} = \mathbf{y}. \tag{1}$$

Se necessario, si può ricorrere alla tecnica del *pivoting*, che consiste nell'effettuare una permutazione delle righe di A. La matrice A viene premoltiplicata per un'oppurtuna matrice di permutazione P. Si ottiene quindi il sistema lineare:

$$PA\mathbf{x} = P\mathbf{b} \Longrightarrow LU\mathbf{x} = P\mathbf{b}.$$

Qualora si utilizzi il pivoting i due sistemi (1) diventano

$$L\mathbf{y} = P\mathbf{b}, \quad U\mathbf{x} = \mathbf{y}. \tag{2}$$

L'algoritmo della fattorizzazione LU con pivoting è il seguente.
Posto $A^{(1)} = A$ (in componenti $a_{ij}^{(1)} = a_{ij}$, per $i, j = 1, \ldots, n$) e $P = I$ si calcoli:

per $k = 1, \ldots, n-1$
  trovare $\bar{r}$ tale che $|a_{\bar{r}k}^{(k)}| = \max\limits_{r=k,\ldots n} |a_{rk}^{(k)}|$
  scambiare la riga k con la riga $\bar{r}$ sia in A che in P
  per $i = k+1, \ldots, n$
    $l_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$,
    per $j = k+1, \ldots, n$
      $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)}$.

Al termine di questo processo, gli elementi della matrice triangolare U sono ottenuti come:

$$u_{ij} = a_{ij} \quad \text{per } i = 1, \ldots, n \quad \text{e} \quad j = 1, \ldots, i,$$

mentre gli elementi di L sono i coefficienti l_{ik} generati dall'algoritmo. In particolare, gli elementi diagonali di L non sono calcolati, perché per l'unicità della fattorizzazione sono posti uguale ad 1.

I sistemi (1) (o la variante con pivotazione (2)) risultano più agevoli da risolvere perché, essendo rispettivamente triangolari inferiore e superiore, possono essere risolti efficientemente con gli schemi delle sostituzioni in avanti e all'indietro. In particolare il sistema $L\mathbf{y} = \mathbf{b}$ può essere risolto con il seguente algoritmo:

$$
\begin{aligned}
y_1 &= \frac{b_1}{l_{11}} \\
y_i &= \frac{1}{l_{ii}}\Big(b_i - \sum_{j=1}^{i-1} l_{ij} y_j\Big), \quad i = 2, \dots, n \quad l_{ii} \neq 0,
\end{aligned} \tag{3}
$$

e in modo analogo $U\mathbf{x} = \mathbf{y}$ con:

$$
\begin{aligned}
x_n &= \frac{y_n}{u_{nn}} \\
x_i &= \frac{1}{u_{ii}}\Big(y_i - \sum_{j=i+1}^{n} u_{ij} x_j\Big), \quad i = n-1, \dots, 1 \quad u_{ii} \neq 0.
\end{aligned} \tag{4}
$$

La funzione Matlab® `lu` calcola la fattorizzazione LU con pivoting per righe. La sua sintassi completa è

```
>>[L,U,P]=lu(A);
```

dove P è la matrice di permutazione.

Esercizio 1

Si vuole risolvere il sistema lineare $A\mathbf{x} = \mathbf{b}$, con A la matrice:

$$
A = \begin{bmatrix}
0.01 & 0 & -0.5 & 0 & \cdots & 0 \\
-1 & 3 & 0 & -0.5 & \cdots & 0 \\
0 & -1 & 3 & 0 & \ddots & \vdots \\
\vdots & 0 & \ddots & \ddots & \ddots & -0.5 \\
\vdots & \vdots & & \ddots & \ddots & 0 \\
0 & 0 & 0 & \cdots & -1 & 3
\end{bmatrix},
$$

e $\mathbf{b}$ il vettore di dimensione n tale per cui la soluzione esatta è $\mathbf{x}_{\text{ex}} = [1, 1, \dots, 1]^T \in \mathbb{R}^n$. Si scelga $n = 50$.

1. Si controlli che il sistema $A\mathbf{x} = \mathbf{b}$ ammette una sola soluzione. Inoltre, si verifichino le condizioni sufficienti e necessaria e sufficiente per l'esistenza e unicità della fattorizzazione LU di A.

2. Si calcoli la fattorizzazione LU con pivoting della matrice A, mediante la funzione Matlab® `lu`. È stata utilizzata la tecnica del pivoting in questo caso?

3. Implementare gli algoritmi di sostituzione in avanti e all'indietro mediante due funzioni Matlab® la cui interfaccia sarà rispettivamente:

   `function [y] = fwsub(L,b)` e `function [x] = bksub(U,y)`

   La funzione Matlab® `fwsub.m`, dati in ingresso una matrice triangolare inferiore $L \in \mathbb{R}^{n\times n}$ e un vettore $\mathbf{b} \in \mathbb{R}^n$, restituisce in uscita il vettore $\mathbf{y}$, soluzione del sistema $L\mathbf{y} = \mathbf{b}$, calcolata mediante l'algoritmo della sostituzione in avanti (3).
   Analogamente, la funzione `bksub.m`, dati in ingresso una matrice triangolare superiore $U \in \mathbb{R}^{n\times n}$ e un vettore $\mathbf{y} \in \mathbb{R}^n$, restituisce in uscita il vettore $\mathbf{x}$, soluzione del sistema $U\mathbf{x} = \mathbf{y}$, calcolata mediante l'algoritmo della sostituzione in indietro (4).

4. Risolvere numericamente, utilizzando le funzioni `fwsub.m` e `bksub.m` implementate al punto precedente, i due sistemi triangolari necessari per ottenere la soluzione del sistema di partenza $A\mathbf{x} = \mathbf{b}$.

Esercizio 2

Spesso, in applicazioni concrete, ci si trova a dover risolvere sistemi lineari la cui matrice è *tridiagonale*, cioè del tipo:

$$A = \begin{bmatrix} a_1 & c_1 & & & \\ e_1 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & e_{n-2} & a_{n-1} & c_{n-1} \\ & & & e_{n-1} & a_n \end{bmatrix}.$$

In tale situazione, un algoritmo molto efficiente è l'*algoritmo di Thomas*:

1. sfrutta la struttura tridiagonale della matrice per calcolare in modo rapido la fattorizzazione LU della matrice A. Le matrici L, U che si ottengono risultano bidiagonali;
2. utilizza tali informazioni sulla struttura di L, U per risolvere efficientemente i due sistemi $L\mathbf{y} = \mathbf{b}$ e $U\mathbf{x} = \mathbf{y}$.

In particolare, se $A \in \mathbb{R}^{n\times n}$ è della forma di cui sopra, allora le matrici L ed U sono date da:

$$L = \begin{bmatrix} 1 & & & & \\ \delta_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & \delta_{n-2} & 1 & \\ & & & \delta_{n-1} & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} \alpha_1 & c_1 & & & \\ & \alpha_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{n-1} & c_{n-1} \\ & & & & \alpha_n \end{bmatrix},$$

con

$$\alpha_1 = a_1, \qquad \delta_{i-1} = \frac{e_{i-1}}{\alpha_{i-1}}, \qquad \alpha_i = a_i - \delta_{i-1}c_{i-1}, \qquad i = 2, \dots, n.$$

Quindi possiamo risolvere in sequenza i due sistemi bidiagonali tramite le relazioni:

$$(L\mathbf{y} = \mathbf{b}) \qquad y_1 = b_1, \qquad y_i = b_i - \delta_{i-1}y_{i-1}, \qquad i = 2, \dots, n$$

$$(U\mathbf{x} = \mathbf{y}) \qquad x_n = \frac{y_n}{\alpha_n}, \qquad x_i = \frac{y_i - c_i x_{i+1}}{\alpha_i}, \qquad i = n-1, \dots, 1.$$

Il costo computazionale complessivo per l'applicazione dell'algoritmo di Thomas è di $8n-7$ operazioni contro le $O(\frac{2}{3}n^3)$ dell'applicazione della fattorizzazione LU.

1. Si implementi in Matlab® l'algoritmo di Thomas per risolvere un sistema lineare tridiagonale. L'interfaccia dovrà essere:

   ```
   function [L,U,x] = thomas(A,b)
   ```

2. Si utilizzi la funzione `thomas` per risolvere il sistema lineare $A\mathbf{x} = \mathbf{b}$, dove $A \in \mathbb{R}^{n\times n}$, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$, per $n = 1000$, con:

   $$A = \texttt{tridiag}(-1, 2-1) \quad \text{e} \quad \mathbf{b} = [1, 1, \dots, 1]^T.$$

   Si riportino i valori di $\mathbf{x}_n$ e $U_{nn} = \alpha_n$ così ottenuti. Si *stimi* inoltre il risparmio computazionale garantito dall'utilizzo dell'algoritmo di Thomas rispetto all'applicazione del metodo di fattorizzazione LU completo.

Esercizio 3 - Homework

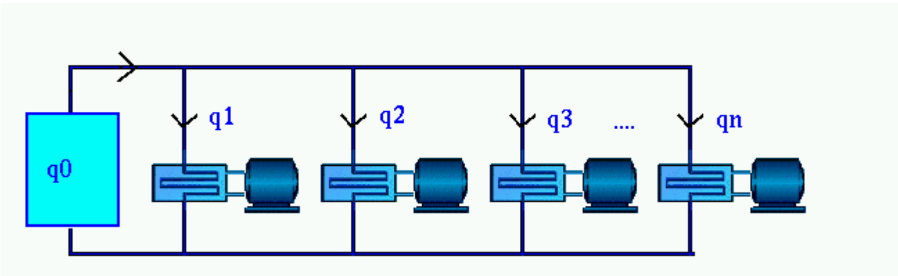
Si considerino le matrici

$$A = \begin{bmatrix} 50 & 1 & 3 \\ 1 & 6 & 0 \\ 3 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 50 & 1 & 10 \\ 3 & 20 & 1 \\ 10 & 4 & 70 \end{bmatrix}, \quad C = \begin{bmatrix} 7 & 8 & 9 \\ 5 & 4 & 3 \\ 1 & 2 & 6 \end{bmatrix}. \tag{5}$$

1. Si verifichi, utilizzando Matlab® o Octave , se le matrici A, B e C soddisfano la condizioni necessaria e sufficiente per l'esistenza della fattorizzazione LU (senza usare pivoting per riga).
2. Utilizzare la funzione `lu` per fattorizzare le matrici A, B, e C.
3. Supponiamo ora di voler risolvere il sistema $A\mathbf{x} = \mathbf{b}$ con A definita in (5). Si utilizzi come termine noto $\mathbf{b}$, un vettore tale che la soluzione esatta del sistema sia $\mathbf{x}_{ex} = [1,\ 1,\ 1]^T$. Si calcoli la soluzione del sistema $A\mathbf{x} = \mathbf{b}$, utilizzando le funzioni `bksub.m` e `fwsub.m`.
4. Si calcoli la norma 2 dell'errore relativo $\|\mathbf{x}_{ex} - \mathbf{x}\|_2/\|\mathbf{x}_{ex}\|_2$ e la norma 2 del residuo normalizzato $\|\mathbf{b} - A\mathbf{x}\|_2/\|\mathbf{b}\|_2$ conoscendo la soluzione esatta.

Esercizio 4 - Homework

Una sorgente di fluido refrigerante di portata q_0 raffredda n macchine distribuite in parallelo come schematizzato in figura.



La caduta di pressione Δp_i in ogni macchina è legata alla portata di fluido q_i che la attraversa tramite la relazione:

$$\Delta p_i = R_i q_i,$$

dove R_i rappresenta la resistenza e gli attriti nel passaggio del fluido attraverso l'i-esima macchina. Si vuole determinare la portata q_i che raggiunge ciascuna macchina. Il calcolo delle portate q_i conduce a un sistema lineare $A\mathbf{q} = \mathbf{b}$, dove $\mathbf{q} = [q_1 \ldots, q_n]^T$ è il vettore delle portate incognite, A è la matrice:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ R_1 & -R_2 & 0 & 0 & \cdots & 0 \\ 0 & R_2 & -R_3 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & R_{n-1} & -R_n \end{bmatrix},$$

e $\mathbf{b}$ è il vettore di dimensione n:

$$\mathbf{b} = [q_0, 0, 0, \cdots, 0]^T.$$

La prima equazione del sistema lineare esprime il fatto che $\sum_i^n q_i = q_0$, mentre le altre $n-1$ equazioni si ricavano tenendo conto che le cadute di pressione Δp_i in ogni macchina sono tutte uguali (essendo le macchine in parallelo), quindi per ogni i, con $i = 1, \ldots, n-1$, possiamo scrivere l'equazione $R_i q_i - R_{i+1} q_{i+1} = 0$.

1. Si ponga $n = 20, R_i = 1$ con $i = 1, \ldots, n$ e $q_0 = 2$ e si assegnino in Matlab® la matrice A e il vettore dei termini noti $\mathbf{b}$.

2. Si calcoli la fattorizzazione LU della matrice A, mediante la funzione Matlab® `lu`. Verificare che la tecnica del pivoting non è stata usata in questo caso.

3. Verificare utilizzando il comando `spy` che la matrice L è sparsa, mentre la matrice U viene riempita.

4. Risolvere numericamente, utilizzando le funzioni `fwsub.m` e `bksub.m`, i due sistemi triangolari necessari per ottenere la soluzione del sistema di partenza $A\mathbf{q} = \mathbf{b}$.

5. Si calcoli la norma 2 dell'errore relativo $\|\mathbf{err_{rel}}\| = \|\mathbf{q}_{ex} - \mathbf{q}\|/\|\mathbf{q}_{ex}\|$ e la norma 2 del residuo normalizzato $\|\mathbf{res_{nor}}\| = \|\mathbf{b} - A\mathbf{q}\|/\|\mathbf{b}\|$ sapendo che la soluzione esatta è il vettore $q_{ex}(i) = \dfrac{q_0}{n}$, $i = 1, \ldots, n$.

6. Si ponga $R_1 = 10^3$ e si calcoli la nuova distribuzione delle portate. dopo aver effettuato la fattorizzazione LU di A. La matrice di pivoting coincide con l'identità? Perché?

Esercizio 5 - Homework

Dato il vettore $\mathbf{p} = [p_1, p_2, \cdots, p_n]^T$, si definisca la matrice di Vandermonde:

$$V(\mathbf{p}) = \begin{bmatrix} p_1^{n-1} & \cdots & p_1^1 & 1 \\ p_2^{n-1} & \cdots & p_2^1 & 1 \\ \vdots & \vdots & \ddots & \vdots \\ p_n^{n-1} & \cdots & p_n^1 & 1 \end{bmatrix}.$$

Fissato n, si costruisca il vettore $\mathbf{p} = [1/n, 2/n, \cdots, 1]^T$ e la matrice V corrispondente (può essere utile utilizzare il comando `vander` di Matlab® che genera la matrice di Vandermonde dato un vettore in ingresso – vedere `help vander`). Si risolva, tramite fattorizzazione LU e metodi di sostituzione in avanti e all'indietro, il sistema $V\mathbf{x} = \mathbf{b}$ per $n = 5$ e con $\mathbf{b}$ scelto in modo tale che $\mathbf{x}_{\text{ex}} = [1, 1, 1, 1, 1]^T$ sia la soluzione esatta.