Risoluzione di un sistema triangolare inferiore

$$\begin{bmatrix} a_{1,1} & 0 & 0 & \dots & 0 & 0 \\ a_{2,1} & a_{2,2} & 0 & \dots & 0 & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3} & \dots & a_{n-1,n-1} & 0 \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n-1} & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

$a_{i,i} \neq 0,\ i = 1, \dots, n.$

Sostituzione in avanti:

$$x_1 = b_1 / a_{1,1}$$

$$\textit{For} \quad i = 2 : n$$

$$x_i = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j \right)$$

Risoluzione di un sistema triangolare superiore

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n-1} & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \dots & a_{2,n-1} & a_{2,n} \\ 0 & 0 & a_{3,3} & \dots & a_{3,n-1} & a_{3,n} \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \dots & 0 & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

$a_{i,i} \neq 0,\ i = 1, \dots, n.$

Sostituzione all'indietro

$$x_n = b_n / a_{n,n}$$

$$Fori = n-1 : -1 : 1$$

$$x_i = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=i+1}^{n} a_{i,j} x_j \right)$$

Esercizi

- Scrivere una funzione di Matlab che implemeti il metodo della sostituzione in avanti.
- Scrivere una funzione di Matlab che implemeti il metodo della sostituzione all'indietro.

Risoluzione di sistemi lineari

Il comando $\backslash$ (*backslash*)

- Se A è una matrice $n \times n$ e $\mathbf{b}$ è un vettore colonna di dimensione n

  `x=A \ b`

  calcola la soluzione del sistema lineare

$$A\,\mathbf{x} = \mathbf{b}\,.$$

- Non calcola l'inversa.
- Risolve il sistema lineare usando l'algoritmo più efficiente, effetuando dei test preliminari sulla matrice: ad esempio, se A è triangolare inferiore usa il metodo della sostituzione in avanti.

Il comando `inv`.

- Calcola esplicitamente l'inversa della matrice A.
- Non è conveninte usare questo comando per risolvere un sistema lineare perche è più costoso.

Altri comandi per matrici (e vettori)

- `det(A)` calcola il determinante della matrice A.
- `v=diag(A)` se A è una matrice calcola il vettore con la diagonale di A.
- `D=diag(v)` se $\mathbf{v}$ calcola la matrice diagonale che ha nella diagonale i valori di $\mathbf{v}$.
- `M=tril(A)` restituisce la parte triangolare inferiore di A. (Provare anche `tril(A,1)` o `tril(A,-2)`).
- `M=triu(A)` restituisce la parte triangolare superiore di A. Analogo a `tril`.
- `v=eig(A)` calcola gli autovalori di A.
- `[M,v]=eig(A)` calcola gli autovettori e gli autovalori di A.

Matrici particolari

```
zeros(3,4), zeros(2)
ones(2,5), ones(3)
eye(4)
```

Osservazione:

$$[c_1, c_2, c_3]\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = c_1 d_1 + c_2 d_2 + c_3 d_3$$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}[d_1, d_2, d_3] = \begin{bmatrix} c_1 d_1 & c_1 d_2 & c_1 d_3 \\ c_2 d_1 & c_2 d_2 & c_2 d_3 \\ c_3 d_1 & c_3 d_2 & c_3 d_3 \end{bmatrix}$$

Metodo di eliminazione di Gauss

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow U\mathbf{x} = \mathbf{c}$$

U matrice triangolare superiore.

for k=1:n-1

  for i=k+1:n

    $m_{i,k} = a_{i,k}^{(k)} / a_{k,k}^{(k)}$

    for j=k+1:n

      $a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - m_{i,k} a_{k,j}^{(k)}$

    end

    $b_i^{(k+1)} = b_i^{(k)} - m_{i,k} b_k^{(k)}$

  end

end

$$U = A^{(n)}, \qquad c = \mathbf{b}^{(n)}.$$

Il comando `lu`

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

- $L\,U = P\,A$
- `L` triangolare inferiore con gli elementi diagonali uguali a 1.
- `U` triangolare superiore.
- `P` matrice di permutazioni.

`[L,U]=lu(A)`

- $L\,U = A$
- `L` non è triangolare inferiore ma una permutazione di righe la rende triangolare inferiore con tutti gli elementi diagonali uguali a 1.
- `U` triangolare superiore.

Il comando `chol`

`R=chol(A)`

- A deve essere simmetrica definita positiva
- $R^T R = A$
- `R` triangolare superiore con tutti gli elementi diagonali positivi.

Risoluzione di un sistema tridiagonale

$$\begin{bmatrix} a_1 & c_1 & & & & \\ b_1 & a_2 & c_2 & & & \\ & b_2 & a_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & b_{n-2} & a_{n-1} & c_{n-1} \\ & & & & b_{n-1} & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}$$

Metodo di Thomas

$$
\begin{bmatrix}
a_1 & c_1 & & & & \\
b_1 & a_2 & c_2 & & & \\
 & b_2 & a_3 & c_3 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & b_{n-2} & a_{n-1} & c_{n-1} \\
 & & & & b_{n-1} & a_n
\end{bmatrix}
$$

$$
= \begin{bmatrix}
1 & & & & & \\
\beta_1 & 1 & & & & \\
 & \beta_2 & 1 & & & \\
 & & \ddots & \ddots & & \\
 & & & \beta_{n-2} & 1 & \\
 & & & & \beta_{n-1} & 1
\end{bmatrix}
\begin{bmatrix}
\alpha_1 & c_1 & & & & \\
 & \alpha_2 & c_2 & & & \\
 & & \alpha_3 & c_3 & & \\
 & & & \ddots & \ddots & \\
 & & & & \alpha_{n-1} & c_{n-1} \\
 & & & & & \alpha_n
\end{bmatrix}
$$

```
alpha(1)=a(1);
for i=1:n-1
  beta(i)=b(i)/alpha(i);
  alpha(i+1)=a(i+1)-beta(i)*c(i);
end
```

Sostituzione in avanti

$$A\mathbf{x} = \mathbf{f} \qquad A = LU \qquad L\mathbf{y} = \mathbf{f} \quad U\mathbf{x} = \mathbf{y}$$

$$\begin{bmatrix} 1 & & & & & \\ \beta_1 & 1 & & & & \\ & \beta_2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & \beta_{n-2} & 1 & \\ & & & & \beta_{n-1} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}$$

```
y(1)=f(1);
for i=2:n
  y(i)=f(i)-beta(i-1)*y(i-1);
end
```

Sostituzione all'indietro

$$\begin{bmatrix} \alpha_1 & c_1 & & & & \\ & \alpha_2 & c_2 & & & \\ & & \alpha_3 & c_3 & & \\ & & & \ddots & \ddots & \\ & & & & \alpha_{n-1} & c_{n-1} \\ & & & & & \alpha_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}$$

```
x(n)=y(n)/alpha(n);
for i=n-1:-1:1
  x(i)=(y(i)-c(i)*x(i+1))/alpha(i);
end
```

Esercizio

Scrivere una funzione di Matlab che implementi l'algoritmo di Thomas.