



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Laboratorio di Calcolo Numerico
Laboratorio 10: Metodi diretti per la soluzione di
sistemi lineari

13 Maggio 2019

Risolvere sistemi di equazioni lineari

Metodi Diretti

Data una matrice quadrata A di ordine n e i vettori colonna \mathbf{x} e \mathbf{b} , si vuole risolvere il seguente sistema di equazioni lineari:

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

con un metodo diretto in cui il numero di passi per giungere alla soluzione \mathbf{x} è fissato.

- Sistemi diagonali e triangolari
- Eliminazione di Gauss
- Fattorizzazione LU (Crout)
- Fattorizzazione di Cholesky

Risolvere sistemi di equazioni lineari

Sistemi DIAGONALI

-
- 1: **per** $i = 1, \dots, n$
 - 2: $x_i = b_i/a_{i,i}$
 - 3: **fine per**
-

In MATLAB:

```
>> A = [1 0 0; 0 5 0; 0 0 8]
```

```
>> b = [12; 33; 36]
```

```
>> x = b./diag(A)
```

```
x =
```

```
12.0000
```

```
6.6000
```

```
4.5000
```

Risolvere sistemi di equazioni lineari

Sistemi TRIANGOLARI

Se la matrice A è triangolare inferiore, il sistema $A\mathbf{x} = \mathbf{b}$ si risolve mediante sostituzioni in avanti:

-
-
- 1: $x_1 = b_1/a_{11}$
 - 2: **per** $i = 2, \dots, n$
 - 3: $x_i = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j) / a_{ii}$
 - 4: **fine per**
-
-

Se la matrice A è triangolare superiore, il sistema $A\mathbf{x} = \mathbf{b}$ si risolve mediante sostituzioni all'indietro:

-
-
- 1: $x_n = b_n/a_{nn}$
 - 2: **per** $i = n - 1, \dots, 1$
 - 3: $x_i = (b_i - \sum_{j=i+1}^n a_{ij}x_j) / a_{ii}$
 - 4: **fine per**
-
-

Risolvere sistemi di equazioni lineari

Sistemi TRIANGOLARI - SOSTITUZIONE IN AVANTI

```
function x=avanti(A,b)
[~,n]=size(A);
x=zeros(n,1);
x(1)=b(1)/A(1,1);
for i=2:n
    somma = 0.;
    for j=1:i-1
        somma = somma + A(i,j)*x(j);
    end
    x(i)=(b(i)- somma)/A(i,i);
end
end
```

Eliminazione di Gauss

Algoritmo

```
1: per  $k = 1, \dots, n$ 
2:   per  $j = k + 1, \dots, n$ 
3:      $u_{kj} = a_{kj}^{(k-1)} / a_{kk}^{(k-1)}$ 
4:   fine per
5:    $c_k = b_k^{(k-1)} / a_{kk}^{(k-1)}$ 
6:   per  $i = k + 1, \dots, n$ 
7:     per  $j = k + 1, \dots, n$ 
8:        $a_{ij}^{(k)} = u_{kj} a_{ik}^{(k-1)} - a_{ij}^{(k-1)}$ 
9:     fine per
10:     $b_i^{(k)} = c_k a_{ik}^{(k-1)} - b_i^{(k-1)}$ 
11:   fine per
12: fine per
13:  $x_n = y_n$ 
14: per  $i = n - 1, \dots, 1$ 
15:    $x_i = y_i - \sum_{j=i+1}^n u_{ij} x_j$ 
16: fine per
```

Eliminazione di Gauss

```
function [U,x]=gauss(A,b)
% GAUSS Soluzione sistema lineare Ax=b con
% eliminazione di Gauss
% [L,x]=gauss(A,b) memorizza in U il fattore
% triangolare alto della fattorizzazione LU
% di A e in x la soluzione di Ax=b

[m, n]=size(A);
if m~=n
    error('La matrice deve essere quadrata')
end
c = zeros(n,1);
U = eye(n);
for k=1:n
    for j=k+1:n
        U(k,j)=A(k,j)/A(k,k);
    end
    c(k)=b(k)/A(k,k);
    for i=k+1:n
        for j=k+1:n
            A(i,j)=U(k,j)*A(i,k)-A(i,j);
        end
        b(i)=c(k)*A(i,k)-b(i);
    end
end
% Sostituzioni all'indietro
x(n) = c(n);
for i=n-1:-1:1
    x(i)=c(i)-(U(i,(i+1):n)*x((i+1):n)');
end
end
```

Fattorizzazione di Crout

- La matrice A viene fattorizzata come $A = LU$ dove L è una triangolare inferiore e U è una triangolare superiore con diagonale unitaria.
- La fattorizzazione triangolare di Crout è un metodo diretto.
- Il sistema lineare $A\mathbf{x} = \mathbf{b}$ può essere scomposto in due sistemi triangolari da risolvere a cascata:

$$L\mathbf{y} = \mathbf{b}$$

$$U\mathbf{x} = \mathbf{y}$$

Fattorizzazione di Crout

Algoritmo

```
1:  $l_{11} = a_{11}$ 
2: per  $j = 2, \dots, n$ 
3:    $u_{1j} = a_{1j}/l_{11}$ 
4: fine per
5: per  $i = 2, \dots, n$ 
6:    $l_{i1} = a_{i1}$ 
7:   per  $j = 2, \dots, i$ 
8:      $l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}$ 
9:   fine per
10:  per  $j = i + 1, \dots, n$ 
11:     $u_{ij} = (a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj})/l_{ii}$ 
12:  fine per
13: fine per
14:  $y_1 = b_1/l_{11}$ 
15: per  $i = 2, \dots, n$ 
16:    $y_i = (b_i - \sum_{j=1}^{i-1} l_{ij}y_j)/l_{ii}$ 
17: fine per
18:  $x_n = y_n$ 
19: per  $i = n - 1, \dots, 1$ 
20:    $x_i = y_i - \sum_{j=i+1}^n u_{ij}x_j$ 
21: fine per
```

Fattorizzazione di Crout

Istruzioni MATLAB

```
function [L,U,x]=crout(A,b)
% CROUT Soluzione sistema lineare Ax=b con
% fattorizzazione LU
% [L,U,x]=crout(A,b) memorizza in L e U il fattore
% triangolare basso e alto della fattorizzazione LU
% di A e in x la soluzione di Ax=b

[m, n]=size(A);
if m~=n
    error('La matrice deve essere quadrata')
end
U = eye(n);
L = zeros(n);
x = zeros(n,1);
y = zeros(n,1);
L(1,1)=A(1,1);
U(1,2:n)=A(1,2:n)/L(1,1);
for i=2:n
    L(i,1)=A(i,1);
    for j=2:i
        L(i,j)=A(i,j)-(L(i,1:j-1)*U(1:j-1,j));
    end
    for j=i+1:n
        U(i,j)=(A(i,j)-(L(i,1:i-1)*U(1:i-1,j)))/L(i,i);
    end
end
... % Sostituzioni in avanti e all'indietro
end
```

Fattorizzazione di Cholesky

Se la matrice A è simmetrica e definita positiva si può applicare la fattorizzazione di Cholesky: $A = LL^T$

Algoritmo

- 1: $l_{11} = \sqrt{a_{11}}$
 - 2: **per** $i = 2, \dots, n$
 - 3: $l_{i1} = a_{i1}/l_{11}$
 - 4: **fine per**
 - 5: **per** $j = 2, \dots, n$
 - 6: $l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}$
 - 7: **per** $i = j + 1, \dots, n$
 - 8: $l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk})/l_{jj}$
 - 9: **fine per**
 - 10: **fine per**
 - 11: $y_1 = b_1/l_{11}$
 - 12: **per** $i = 2, \dots, n$
 - 13: $y_i = (b_i - \sum_{j=1}^{i-1} l_{ij}y_j)/l_{ii}$
 - 14: **fine per**
 - 15: $x_n = y_n/l_{nn}$
 - 16: **per** $i = n - 1, \dots, 1$
 - 17: $x_i = (y_i - \sum_{j=i+1}^n l_{ji}x_j)/l_{ii}$
 - 18: **fine per**
-

Fattorizzazione di Cholesky

ESERCIZIO - Istruzioni MATLAB

```
function [L,x]=cholesky(A,b)
% CHOLESKY Fattorizzazione  $LL^T$  della
% matrice A per la soluzione del sistema lineare  $Ax=b$ 
% [L,x]=cholesky(A) restituisce in L il fattore
% triangolare basso della fattorizzazione  $LL^T$  e in
% x la soluzione di  $Ax=b$ 

[m, n]=size(A);
if m~=n
    error('La matrice deve essere quadrata')
end
lambda=eig(A);
if any(imag(lambda)~=0 | (real(lambda)<0))
    error('La matrice deve essere definita positiva')
end
...           % Inizializzo L
...           % Elemento (1,1) di L
...           % Prima colonna di L
for j=2:n
    ...
end
...           % Sostituzioni Avanti
...           % Sostituzioni Indietro
```

Funzioni utili di MATLAB

Funzioni MATLAB predefinite per la fattorizzazione LU di una matrice A .

<code>>> [L,U] = lu(A)</code>	restituisce le matrici triangolare inferiore L e superiore U ottenute da fattorizzazione di Doolittle (se non sono necessarie permutazioni). Se ci sono scambi di righe, restituisce la matrice triangolare superiore U mentre in L restituisce il prodotto di matrici triangolari inferiori e matrici di permutazione
<code>>> [L,U,P] = lu(A)</code>	L e U sono triangolare inferiore e superiore mentre P è la matrice di permutazione tale che $PA = LU$
<code>>> U=chol(A)</code>	function che esegue la fattorizzazione di Choleshy. In U viene restituita la matrice triangolare superiore tale che $U^T U = A$
<code>>> L=chol(A,'lower')</code>	function che esegue la fattorizzazione di Choleshy. In L viene restituita la matrice triangolare inferiore tale che $LL^T = A$

L'operatore \

Data una matrice quadrata A di ordine n e i vettori colonna \mathbf{x} e \mathbf{b} , si vuole risolvere il seguente sistema di equazioni lineari:

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

con un metodo diretto in cui il numero di passi per giungere alla soluzione \mathbf{x} è fissato.

- In MATLAB, l'operatore di divisione \ (backslash) permette di risolvere il sistema lineare $A\mathbf{x} = \mathbf{b}$
- Se \mathbf{b} è il vettore termine noto memorizzato come vettore colonna allora l'istruzione $\mathbf{x} = A \setminus \mathbf{b}$ restituisce in \mathbf{x} la soluzione del sistema lineare.
- Il metodo applicato per risolvere il sistema dipende dalla tipologia della matrice A .
- L'operatore \ si basa su metodi diretti.

L'operatore \

ESEMPIO

```
>> A = [1 2 3; 4 5 6; 7 8 0]
```

```
>> b = [12; 33; 36]
```

```
>> x = A \ b
```

```
x =
```

```
4.0000
```

```
1.0000
```

```
2.0000
```

ESERCIZI

- 1 Implementare la function `sindietro` che data una matrice A triangolare superiore e un vettore \mathbf{b} risolva il sistema $A\mathbf{x} = \mathbf{b}$ con il metodo di sostituzione all'indietro. Utilizzare la function per risolvere $A\mathbf{x} = \mathbf{b}$ con A :

$$A = \begin{pmatrix} 2 & 6 & 4 \\ 0 & -3 & 0 \\ 0 & 0 & -3 \end{pmatrix}$$

e $\mathbf{b} = [10, -6, 3]^T$.

- 2 Data la matrice A :

$$A = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 3 \end{pmatrix}$$

fattorizzare secondo Choleky $A = LL^T$ e risolvere il sistema $A\mathbf{x} = \mathbf{b}$ con $\mathbf{b} = [1, 1, 1]^T$ mediante i due sistemi triangolari:

$$L\mathbf{y} = \mathbf{b}$$

$$L^T\mathbf{x} = \mathbf{y}$$

con sostituzioni in avanti e all'indietro.

ESERCIZI

- 3 Data la matrice A :

$$A = \begin{pmatrix} 4 & 2 & 1 & 1 \\ 3 & 2 & 4 & 6 \\ 18 & 9 & 3 & 2 \\ 2 & 4 & 6 & 10 \end{pmatrix}$$

fattorizzare usando la function `lu`. Risolvere il sistema lineare $A\mathbf{x} = \mathbf{b}$ con $\mathbf{b} = [8, 15, 32, 22]^T$ mediante sostituzioni in avanti e all'indietro (permutare in modo opportuno il termine noto \mathbf{b}).

- 4 Fattorizzare secondo Cholesky la matrice A :

$$A = \begin{pmatrix} 16 & -8 & 4 \\ -8 & 20 & 4 \\ 4 & 4 & 12.25 \end{pmatrix}$$

Sfruttare la fattorizzazione per verificare che $\det(A) = \det(L)^2$.