

## RELAZIONE SETTIMA ESERCITAZIONE

Implementare i metodi di Jacobi e Gauss-Seidel (sia in formato matriciale che in formato sparso) per risolvere sistemi lineari aventi come matrice dei coefficienti le matrici ottenuti con i seguenti comandi matlab:

```
[c,d,e]=gallery('dorr',n,0.1);  
A1=gallery('tridiag',c,d,e);  
A1=full(A1);
```

e

```
A2 = gallery('frank',n,2)
```

e termine noto scelto in maniera tale che la soluzione del sistema lineare sia il vettore unitario.

Fissata una tolleranza, calcolare la soluzione al variare dell'ordine della matrice, calcolare il raggio spettrale della matrice di iterazione, l'indice di condizionamento della matrice del sistema lineare, indicare il numero di passi necessari per ottenere la soluzione, il tempo di calcolo.

Giustificare i risultati alla luce della teoria. Confrontare i tempi di risoluzione dei metodi in formato matriciale ed in formato sparso.

## CODICE MATLAB, ANALISI E RISULTATI:

```
function [ x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(n,metodo,tipo)
    x1 = 0;
    c1 = 0;
    r1 = 0;
    k1 = 0;
    t1 = 0;
    x2 = 0;
    c2 = 0;
    r2 = 0;
    k2 = 0;
    t2 = 0;
%matrice 1
[c,d,e]=gallery('dorr',n,0.1);
A1=gallery('tridiag',c,d,e);
A1=full(A1);
%matrice 2
A2 = gallery('frank',n,2);

%Matrice A1
%termine noto = somma delle righe della matrice
b=sum(A1,2); %somma per righe la matrice A1 (la soluzione è: tutte
le x = 1)
%errore (criterio di arresto)
err = 1e-6;
%calcolo E,D,F
F = zeros(n,n) + triu(A1,1); %parte superiore
E = zeros(n,n) + tril(A1,-1); %parte inferiore
D = A1 - triu(A1,1) - tril(A1,-1); %diagonale
%vettore iniziale
xiniziale = zeros(n,1);
%controllo come devo calcolare la matrice di iterazione
if (strcmp(metodo,'jacobi'))
    %JACOBI per matrice 1
    M = D;
    N = -(E+F);
elseif (strcmp(metodo,'gauss_seidel'))
    %GAUSS_SEIDEL per matrice 1
    M = E + D;
    N = -F;
else
    disp('input non valido');
    return
end
T = M\N; %M(-1)*N, matrice di iterazione
%calcolo raggio spettrale della matrice di iterazione
r1 = max(abs(eig(T))); %eig trova gli autovalori, abs fa il valore
assoluto, max prende il massimo dei valori assoluti del raggio
spettrale
%calcolo indice di condizionamento della matrice del sistema
lineare
```

```

c1 = cond(A1);
if (r1 > 1)
    disp('il raggio spettrale della matrice di iterazione del
primo sistema è maggiore di 1, quindi il metodo non converge');
    return
end
[x1,k1,t1] = iterativo(M,N,b,xiniziale,err,tipo);

%Matrice A2
%calcolo E,D,F
F = zeros(n,n) + triu(A2,1); %parte superiore
E = zeros(n,n) + tril(A2,-1); %parte inferiore
D = A2 - triu(A2,1) - tril(A2,-1); %diagonale
%calcolo termine noto = somma delle righe della matrice
b=sum(A2,2); %somma per righe la matrice A1 (la soluzione è: tutte
le x = 1)
%vettore iniziale
xiniziale = zeros(n,1);
%controllo come devo calcolare la matrice di iterazione
if (strcmp(metodo,'jacobi'))
    %JACOBI per matrice 1
    %calcolo raggio spettrale della matrice di iterazione
    M = D;
    N = -(E+F);
elseif (strcmp(metodo,'gauss_seidel'))
    %GAUSS_SEIDEL per matrice 1
    M = E + D;
    N = -F;
else
    disp('input non valido');
    return
end
T = M\N; %M(-1)*N, matrice di iterazione
%calcolo raggio spettrale della matrice di iterazione
r2 = max(abs(eig(T))); %eig trova gli autovalori, abs fa il valore
assoluto, max prende il massimo dei valori assoluti del raggio
spettrale
%calcolo indice di condizionamento della matrice del sistema
lineare
c2 = cond(A2);
if (r2 > 1)
    disp('il raggio spettrale della matrice di iterazione del
secondo sistema è maggiore di 1, quindi il metodo non converge');
    return
end
[x2,k2,t2] = iterativo(M,N,b,xiniziale,err,tipo);

end

```

```

function [ x,k,t ] = iterativo(M,N,b,xiniziale,err,tipo)
%k parte da zero
%la x-iniziale è la  $x^{k-1}$ , x è la  $x^k$ 
%inizializzo vettore di output
x = zeros(length(b),1);
%trovo matrice di iterazione
T = M\N;
if (strcmp(tipo,'piene'))
    x = T*xiniziale + M\b;
else
    %trovo i 3 vettori che rappresentano la matrice in formato
    sparso
    [r,c,i] = vettori_CRS(T);
    %prodotto in forma sparso
    x = (sparsa(r,c,i,xiniziale))' + M\b; %sparsa ritorna il
    vettore scritto per riga, deve essere scritto per colonna
end

% aggiorno k
k = 1;
%calcolo errore
errore = norm(x - xiniziale);
tic
while(errore > err)
    %eseguo gli stessi passi finchè l'errore è minore dell'errore
    massimo
    %(err-max)
    %aggiorno il vettore iniziale
    xiniziale = x;
    %calcolo la nuova  $x^k$ 
    if (strcmp(tipo,'piene'))
        x = T*xiniziale + M\b;
    else
        %trovo i 3 vettori che rappresentano la matrice in formato
        sparso
        [r,c,i] = vettori_CRS(T);
        %prodotto in forma sparso
        x = (sparsa(r,c,i,xiniziale))' + M\b; %sparsa ritorna il
        vettore scritto per riga, deve essere scritto per colonna
    end
    %calcolo il nuovo errore
    errore = norm(x - xiniziale);
    %aggiorno k
    k = k + 1;
end
t = toc;
end

```

```

function [ prod ] = sparsa( r, c, i , vet )
% vettore risultato, bisogna controllare col vettore i se bisogna
%fare
% eventuali somme tra gli elementi del vettore tmp
tmp = r.*vet(c);
count = 1;
% vettore che contiene il risultato finale
prod = zeros(1,length(i)-1);
% scorre il vettore i
for k=1:1:length(i)-1
stmp = 0;
% controlla se la differenza dei 2 elementi consecutivi è > 1
if i(k+1) - i(k) > 1
% somma tanti elementi quant'è la differenza
for j=count:1:count + i(k+1)-i(k) - 1
stmp = stmp + tmp(j);
end
count = count + i(k+1)-i(k);
prod(k) = stmp;
else
prod(k) = tmp(count);
count = count + 1;
end
end
end

```

```

function [ r,c,i ] = vettori_CRS(A)
[I,J,vals] = find(A'); %A' è la trasposta
r = vals; %valori non nulli visti per righe
c = I;
temp=(A~=0); %trova elementi non nulli e ci mette 1 (temp è una
matrice)
d = sum(temp,2); %trova quanti elementi non nulli sono in una riga
(è un vettore)
i = [1;cumsum(d) + 1]; %somma cumulativa
% i(2:end+1) = i; %trasla gli elementi di 1 posizione in avanti
% i(1) = 1; %il vettore i è completo
end

```

## RISULTATI:

```
[ x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(5,'jacobi','piene')
```

il raggio spettrale della matrice di iterazione del secondo sistema è maggiore di 1, quindi il metodo non converge

x1 = (soluzione con matrice A1)

```
1.0000
1.0000
1.0000
1.0000
1.0000
```

c1 = 22.2209 (indice di condizionamento)

r1 = 0.9101 (raggio spettrale)

k1 = 134 (numero di passi)

t1 = 0.0156 (tempo di esecuzione)

x2 = (soluzione con matrice A2)

```
0
```

c2 = 647.4683 (indice di condizionamento)

r2 = 1.8019 (raggio spettrale)

k2 = 0 (numero di passi)

t2 = 0 (tempo di esecuzione)

```
[ x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(5,'jacobi','sparse')
```

il raggio spettrale della matrice di iterazione del secondo sistema è maggiore di 1, quindi il metodo non converge

```
x1 = (soluzione con matrice A1)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c1 = 22.2209 (indice di condizionamento)
```

```
r1 = 0.9101 (raggio spettrale)
```

```
k1 = 134 (numero di passi)
```

```
t1 = 0.0525 (tempo di esecuzione)
```

```
x2 = (soluzione con matrice A2)
```

```
0
```

```
c2 = 647.4683 (indice di condizionamento)
```

```
r2 = 1.8019 (raggio spettrale)
```

```
k2 = 0 (numero di passi)
```

```
t2 = 0 (tempo di esecuzione)
```

```
[ x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(5,'gauss_seidel','piene')
```

```
x1 = (soluzione con matrice A1)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c1 = 22.2209 (indice di condizionamento)
```

```
r1 = 0.8282 (raggio spettrale)
```

```
k1 = 70 (numero di passi)
```

```
t1 = 0.0080 (tempo di esecuzione)
```

```
x2 = (soluzione con matrice A2)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c2 = 647.4683 (indice di condizionamento)
```

```
r2 = 0.9638 (raggio spettrale)
```

```
k2 = 272 (numero di passi)
```

```
t2 = 0.0234 (tempo di esecuzione)
```



```
[ x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(5,'gauss_seidel','sparse')
```

```
x1 = (soluzione con matrice A1)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c1 = 22.2209 (indice di condizionamento)
```

```
r1 = 0.8282 (raggio spettrale)
```

```
k1 = 70 (numero di passi)
```

```
t1 = 0.0134 (tempo di esecuzione)
```

```
x2 = (soluzione con matrice A2)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c2 = 647.4683 (indice di condizionamento)
```

```
r2 = 0.9638 (raggio spettrale)
```

```
k2 = 272 (numero di passi)
```

```
t2 = 0.0555 (tempo di esecuzione)
```

Da questa prima prova si vede come lo stesso sistema lineare possa essere risolto con un metodo iterativo ma non con un altro.

L'esempio si vede col sistema lineare formato dalla matrice A2, il metodo di Jacobi non converge perchè la matrice di iterazione ha raggio spettrale maggiore di 1, quindi non può essere usato questo metodo per trovare la soluzione del sistema lineare.

Col metodo di Gauss\_Seidel invece la soluzione si può trovare, perchè la matrice di iterazione ha raggio spettrale minore di 1, quindi il metodo converge.

Nel primo sistema lineare inoltre si vede che il metodo di Jacobi trova la soluzione con un numero di passi maggiore rispetto al metodo di Gauss-Seidel (134 passi - 70 passi).

Un'altra cosa che si vede è la differenza del tempo di esecuzione dello stesso metodo ma con matrici piene / sparse.

Per entrambi i metodi, la risoluzione con matrici piene è più veloce rispetto alla risoluzione con matrici sparse.

Ora viene fatta un'altra prova, ma con matrici più grandi.

```
[ x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(15,'jacobi','piene')
```

il raggio spettrale della matrice di iterazione del secondo sistema è maggiore di 1, quindi il metodo non converge

```
x1 = (soluzione con matrice A1)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c1 = 175.7708 (indice di condizionamento)
```

```
r1 = 0.9883 (raggio spettrale)
```

```
k1 = 933 (numero di passi)
```

```
t1 = 0.0566 (tempo di esecuzione)
```

```
x2 = (soluzione con matrice A2)
```

```
0
```

```
c2 = 1.3710e+013 (indice di condizionamento)
```

```
r2 = 2.6288 (raggio spettrale)
```

```
k2 = 0 (numero di passi)
```

```
t2 = 0 (tempo di esecuzione)
```

```
[ x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(15,'jacobi','sparse')
```

il raggio spettrale della matrice di iterazione del secondo sistema è maggiore di 1, quindi il metodo non converge

```
x1 = (soluzione con matrice A1)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c1 = 175.7708 (indice di condizionamento)
```

```
r1 = 0.9883 (raggio spettrale)
```

```
k1 = 933 (numero di passi)
```

```
t1 = 0.3597 (tempo di esecuzione)
```

```
x2 = (soluzione con matrice A2)
```

```
0
```

```
c2 = 1.3710e+013 (indice di condizionamento)
```

```
r2 = 2.6288 (raggio spettrale)
```

```
k2 = 0 (numero di passi)
```

```
t2 = 0 (tempo di esecuzione)
```

```
[ x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(15,'gauss_seidel','piene')
```

```
x1 = (soluzione con matrice A1)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c1 = 175.7708 (indice di condizionamento)
```

```
r1 = 0.9767 (raggio spettrale)
```

```
k1 = 482 (numero di passi)
```

```
t1 = 0.0551 (tempo di esecuzione)
```

```
x2 = (soluzione con matrice A2)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
0.9996  
1.0004
```

```
c2 = 1.3710e+013 (indice di condizionamento)
```

r2 = 0.9980 (raggio spettrale)

k2 = 13517 (numero di passi)

t2 = 1.0577 (tempo di esecuzione)

```
[x1,c1,r1,k1,t1,x2,c2,r2,k2,t2 ] = es7(15,'gauss_seidel','sparse')
```

```
x1 = (soluzione con matrice A1)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000
```

```
c1 = 175.7708 (indice di condizionamento)
```

```
r1 = 0.9767 (raggio spettrale)
```

```
k1 = 482 (numero di passi)
```

```
t1 = 0.1241 (tempo di esecuzione)
```

```
x2 = (soluzione con matrice A2)
```

```
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
1.0000  
0.9996  
1.0004
```

```
c2 = 1.3710e+013 (indice di condizionamento)
```

r2 = 0.9980 (raggio spettrale)

k2 = 13517 (numero di passi)

t2 = 3.2473 (tempo di esecuzione)

Anche con matrici più grandi si mantiene l'andamento che si è visto con matrici più piccole.

Infatti:

Il metodo di Jacobi non converge per la risoluzione del secondo sistema lineare, mentre il metodo di Gauss\_Seidel sì;

Il metodo di Jacobi trova la soluzione del primo sistema lineare con un numero di passi maggiore rispetto al metodo di Gauss-Seidel (933 passi - 482 passi);

L'esecuzione del metodo con matrici piene è più veloce rispetto all'esecuzione con matrici sparse (sia per Jacobi che per Gauss\_Seidel).