

GRADIENTE CONIUGATO PRECONDIZIONATO IN MATLAB

MANOLO VENTURIN

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIP. MATEMATICA PURA ED APPLICATA

2008

Problema

- Obiettivo

- Risoluzione del sistema lineare $\mathbf{Ax} = \mathbf{b}$.

- Metodo risolutivo

- Metodo del Gradiente
 - Metodo del Gradiente Coniugato
 - Gradiente coniugato preconditionato

Ideale per sistemi lineari di grandi dimensioni oppure quando è noto un buon punto iniziale per il metodo.

- Requisiti del metodo risolutivo

- \mathbf{A} matrice sparsa simmetrica e definita positiva
 - \mathbf{b} vettore dei termini noti
 - Opportuno preconditionatore \mathbf{P} (simm. e def. pos.)
risolve $\mathbf{P}^{-1}\mathbf{Ax} = \mathbf{P}^{-1}\mathbf{b}$ come
 $\mathbf{P}^{-1/2}\mathbf{AP}^{-1/2}\mathbf{y} = \mathbf{P}^{-1/2}\mathbf{b}$ con $\mathbf{y} = \mathbf{P}^{1/2}\mathbf{x}$
 - Precondizionatori disponibili
(Diagonale e Cholesky incompleto)

Gradiente

```
function [x,xhist] = gradiente(A,b,x0,toll,maxit)
%GRADIENTE Implementa il metodo del gradiente
% SINTASSI
%     [x,xhist] = gradiente(A,b,x0,toll,maxit)
% INPUT
%     A       : Matrice simmetrica e definita positiva
%     b       : Vettore dei termini noti
%     x0      : Punto iniziale
%     toll    : Tolleranza del metodo
%     maxit   : Numero massimo di iterazioni
% OUTPUT
%     x       : ultimo punto trovato
%     xhist   : matrice delle successioni

% Autore : M. Venturin
```

Gradiente

```
x = x0(:);           % punto corrente
n = length(x);       % dimensione
iter = 1;            % iterazione corrente
xhist = zeros(n,maxit); % Allocazione vettore uscita
xhist(:,iter) = x;
r = b-A*x0;          % residuo iniziale
bnrm2 = norm(b);      % norma iniziale
if bnrm2 < toll
    bnrm2 = 1.0;
end
errore = norm(r);     % errore iniziale
```

Gradiente

```
% ciclo principale
while errore > bnorm2*toll && iter<maxit
    iter = iter+1;           % incr. iterazione
    z = A*r;                 % z vett. ausiliario
    alpha = r'*r/(z'*r);    % alpha
    x = x+alpha*r;           % x nuovo
    if mod(iter,50)==0
        r = b-A*x;          % nuovo residuo
    else
        r = r-alpha*z;      % nuovo residuo
    end
    errore = norm(r);        % nuovo errore
    xhist(:,iter) = x;       % salvataggio x
end
xhist = xhist(:,1:iter);
```

Gradiente coniugato

```
function [x,xhist] = gradcon(A,b,x0,toll,maxit)
%GRADIENTE Implementa il metodo del gradiente coniugato
% SINTASSI
%     [x,xhist] = gradcon(A,b,x0,toll,maxit)
% INPUT
%     A      : Matrice simmetrica e definita positiva
%     b      : Vettore dei termini noti
%     x0     : Punto iniziale
%     toll   : Tolleranza del metodo
%     maxit  : Numero massimo di iterazioni
% OUTPUT
%     x      : ultimo punto trovato
%     xhist  : matrice delle successioni

% Autore : M. Venturin
```

Gradiente coniugato

```
x = x0(:);           % punto corrente
n = length(x);       % dimensione
iter = 1;            % iterazione corrente
xhist = zeros(n,maxit); % Allocazione vettore uscita
xhist(:,iter) = x;
r = b-A*x0;          % residuo iniziale
p = r;
bnrm2 = norm(b);      % norma iniziale
if bnrm2 < toll
    bnrm2 = 1.0;
end
errore = norm(r);     % errore iniziale
```

Gradiente coniugato

```
% ciclo principale
while errore > bnorm2*toll && iter<maxit
    iter = iter+1;           % iterazione
    w = A*p;                 % w vett. ausiliario
    alpha = -r'*p/(p'*w); % alpha
    x = x-alpha*p;           % nuova x
    if mod(iter,50)==0
        r = b-A*x;           % nuovo residuo
    else
        r = r+alpha*w;       % nuovo residuo
    end
    p = r-((p'*A*r)/(p'*w))*p; % p
    errore = norm(r);         % nuovo errore
    xhist(:,iter) = x;        % salvataggio x
end
xhist = xhist(:,1:iter);
```


Gradiente coniugato

```
% FILE: testGradienti.m
% Esegue il test 2D del
%   o metodo del gradiente, e
%   o metodo del gradiente coniugato

% pulizia variabili, schermo
close all; clear all; clc

% sistema lineare simm. e def. pos.
A = [8 4; 4 3]; xsol = [-2;6]; b = A*xsol;

% Punto iniziale
x0 = [1;0];

% Metodo del gradiente
[xg,xghist] = gradiente(A,b,x0,1e-8,100);

% Metodo del gradiente coniugato
[xc,xchist] = gradcon(A,b,x0,1e-8,100);
```

Gradiente coniugato

```
% Disegno andamento della soluzione
funz = 'Z = 4.*X.^2+4.*X.*Y+3./2.*Y.^2-8.*X-10.*Y;';
X = xghist(1,:); Y = xghist(2,:); eval(funz); levels = Z;
[X,Y] = meshgrid(-6:0.1:2,-1:0.1:12); eval(funz);
figure;
contour(X,Y,Z,levels,'k');
hold on;
for i=1:size(xghist,2)-1
    line([xghist(1,i),xghist(1,i+1)],...
        [xghist(2,i),xghist(2,i+1)]);
    plot(xghist(1,i),xghist(2,i),'b*');
end
plot(xsol(1),xsol(2),'r*');
hold off;
title('metodo del gradiente');
```

Gradiente conjugato

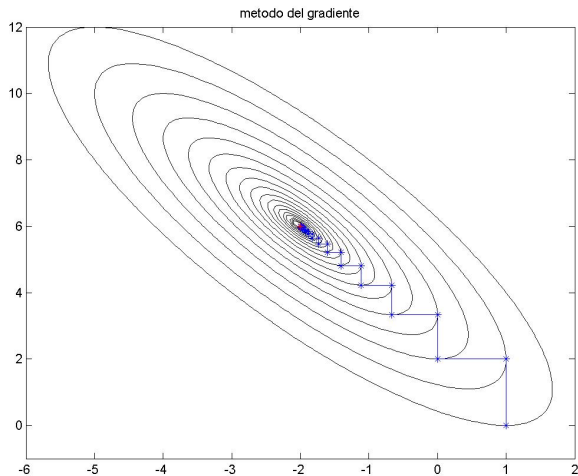


Figura: Andamento del metodo del Gradiente

Gradiente coniugato

```
% Disegno andamento della soluzione
funz = 'Z = 4.*X.^2+4.*X.*Y+3./2.*Y.^2-8.*X-10.*Y;';
X = xhist(1,:); Y = xhist(2,:); eval(funz); levels = Z;
[X,Y] = meshgrid(-6:0.1:2,-1:0.1:12); eval(funz);
figure;
contour(X,Y,Z,levels,'k');
hold on;
for i=1:size(xhist,2)-1
    line([xhist(1,i),xhist(1,i+1)],...
        [xhist(2,i),xhist(2,i+1)]);
    plot(xhist(1,i),xhist(2,i),'b*');
end
plot(xsol(1),xsol(2),'r*');
hold off;
title('metodo del gradiente coniugato');
```

Gradiente conjugato

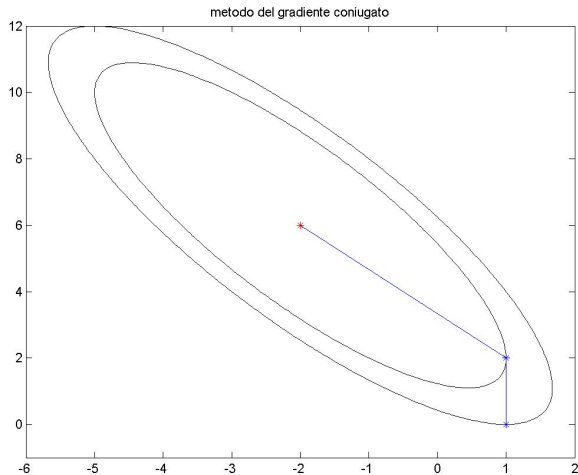


Figura: Andamento del metodo del Gradiente

PCG di MatLab

- ▶ PCG implementa il gradiente coniugato preconditionato
- ▶ Sintassi utilizzata in questi esempi

$$[X, FLAG, RELRES, ITER, RESVEC] =$$
$$PCG(A, B, TOL, MAXIT, M1, M2, X0, P1, P2)$$

Nota:

- ▶ Il preconditionatore P è della forma $P = M1 \times M2$ dove $M1$ ed $M2$ sono due sue fattorizzazioni.

PCG di MatLab - Esempio

- ▶ Problema di Laplace
 - ▶ $-\Delta u = 1$ nel quadrato unitario $\Omega = (0, 1) \times (0, 1)$
 - ▶ $u = 0$ sul bordo del dominio $\partial\Omega$
- ▶ Discretizzazione degli operatori
 - ▶ metodo alle differenze finite
- ▶ Risoluzione del sistema lineare ottenuto mediante
 - ▶ metodo diretto di eliminazione gaussiana (confronto)
 - ▶ metodo del gradiente coniugato preconditionato
- ▶ Visualizzazione dei risultati
 - ▶ Griglia
 - ▶ Pattern di sparsità della matrice **A**
 - ▶ Soluzione
 - ▶ Residuo relativo dell'andamento del metodo PCG
 - ▶ Analisi spettrale del sistema e del sistema preconditionato

Discretizzazione del Problema

```
% FILE : pcgesempio.m

close all; clear all; clc;
%%% COSTRUZIONE DEL SISTEMA LINEARE
% Problema :  $-\nabla^2(u) = 1$  in un quadrato unitario
%               $u = 0$  sul bordo

% Generazione e visualizzazione della griglia
n = 24; G = numgrid('S',n);
figure; spy(G); title('Nodi della griglia');

% Discretizzazione operatore Laplaciano
% nei nodi interni
A = delsq(G);
figure; spy(A); title('Matrice sistema');

% Calcolo vettore dei termini noti nei nodi interni
N = sum(G(:)>0); % numero nodi interni
b = ones(N,1); % vettore termini noti
% Sistema lineare da risolvere :  $A x = b$  (nodi interni)
```


Discretizzazione del Problema

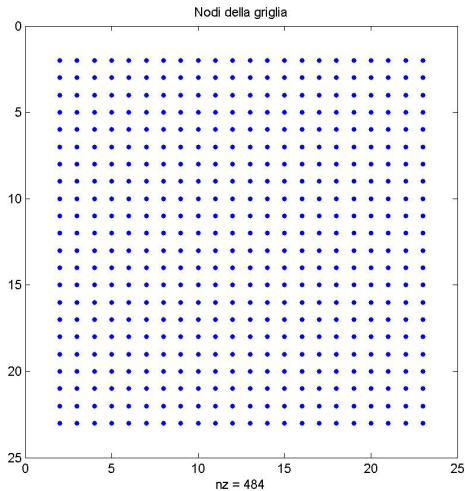


Figura: Griglia di calcolo

Discretizzazione del Problema

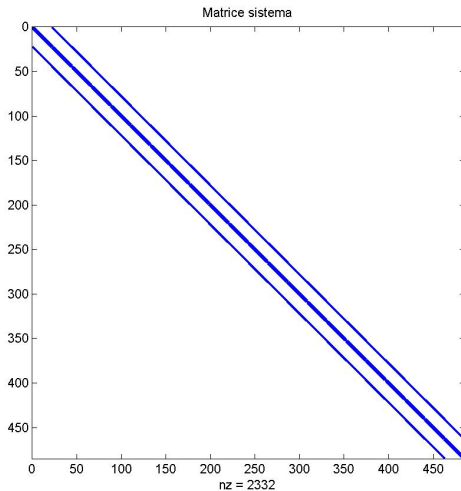


Figura: Pattern di sparsità della matrice

Risoluzione: Metodo diretto

```
%%% RISOLUZIONE CON METODO DIRETTO
% Risoluzione del sistema  $A x = b$  associato
% al problema di Laplace

% Chiamata al metodo diretto
u = A\b;

% Visualizzazione curve di livello soluzione
U = G; U(G>0) = full(u(G(G>0)));
figure; clabel(contourf(U)); prism; axis square ij

% Visualizzazione della soluzione
figure; mesh(U);
axis([0 n 0 n 0 max(max(U))]); axis square ij

% Residuo soluzione (6.9941e-013)
resd = norm(A*u-b);
```

Risoluzione: Metodo diretto

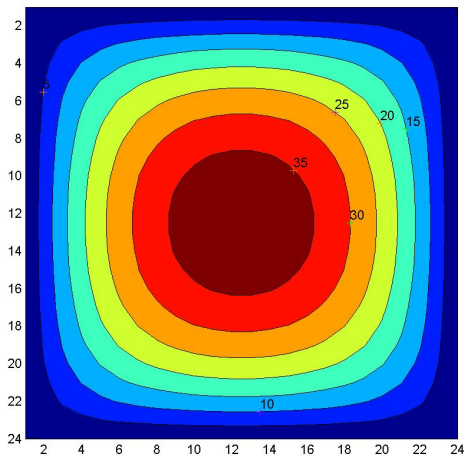


Figura: Curve di livello della soluzione

Risoluzione: Metodo diretto

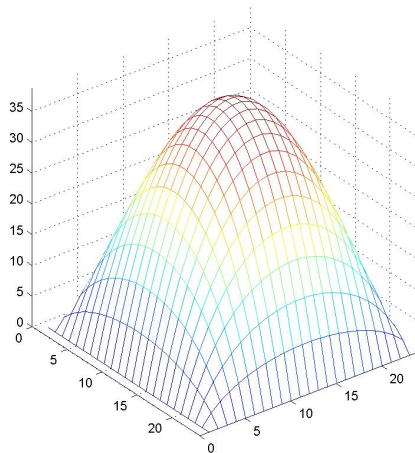


Figura: Soluzione calcolato mediante metodo diretto

Risoluzione: Gradiente coniugato prec. diag.

```
% PCG CON PRECONDIZIONATORE DIAGONALE

% Calcolo del preconditionatore
D = spdiags(sqrt(spdiags(A,0)),0,size(A,1),size(A,2));

% Calcolo della soluzione
[upcg1,flag1,relres1,iter1,resvec1] = pcg(A,b,1e-8,50,D',D);

% Visualizzo andamento della soluzione
figure; h = semilogy(0:iter1,resvec1/norm(b),'-o');
set(h,'Linewidth',2);
xlabel('numero iterazione'); ylabel('residuo relativo');
grid on;
```

Risoluzione: Gradiente coniugato prec. diag.

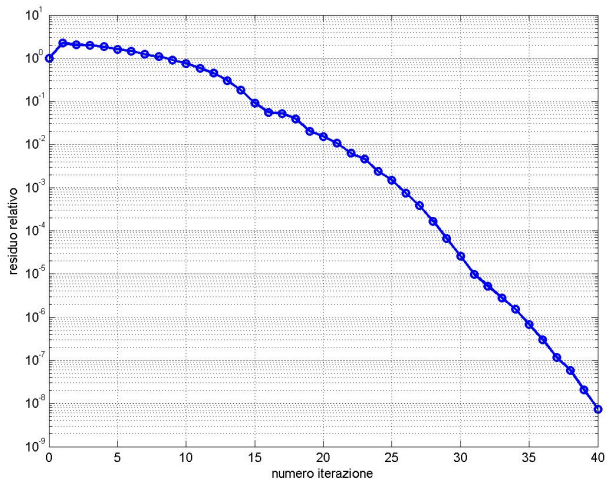


Figura: Andamento del residuo relativo nel metodo pcg - prec. diagonale

Risoluzione: Gradiente coniugato prec. cholinc

```
% PCG CON PRECONDIZIONATORE INCOMPLETO DI CHOLESKY

% Calcolo del preconditionatore
R = cholinc(A,1e-3);

% Calcolo della soluzione
[upcg2,flag2,relres2,iter2,resvec2] = pcg(A,b,1e-8,50,R',R);

% Visualizzo andamento della soluzione
figure; h = semilogy(0:iter2,resvec2/norm(b),'-o');
set(h,'Linewidth',2);
xlabel('numero iterazione'); ylabel('residuo relativo');
grid on;

% Pattern del preconditionatore
figure; spy(R'*R);
```


Risoluzione: Gradiente coniugato prec. cholinc

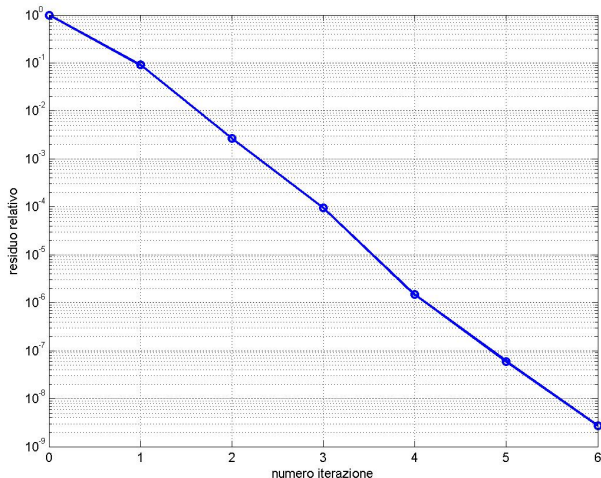


Figura: Andamento del residuo relativo nel metodo pcg - prec. Cholesky incompleto

Risoluzione: Gradiente coniugato prec. cholinc

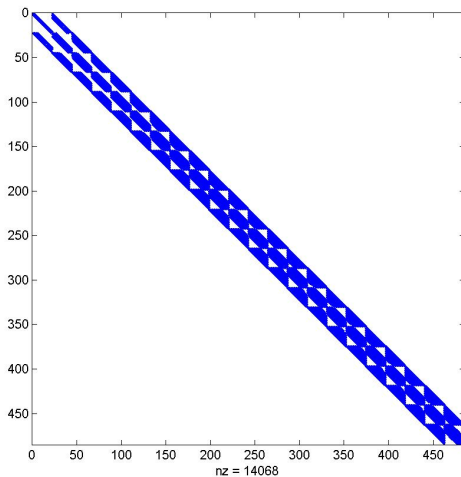


Figura: Pattern del preconditionatore di Cholesky incompleto

Analisi spettrale

```
% ANALISI SPETTRALE

% Autovalori del sistema originario
eigo = eig(full(A));
% Visualizzazione del relativo spettro
figure; plot(real(eigo),imag(eigo),'xk');

% Autovalori del sistema preconditionato diag
eigp1 = eig(inv(full(D'))*full(A)*inv(full(D)));
% Visualizzazione del relativo spettro
figure; plot(real(eigp1),imag(eigp1),'xk');

% Autovalori del sistema preconditionato cholinc
eigp2 = eig(inv(full(R'))*full(A)*inv(full(R)));
% Visualizzazione del relativo spettro
figure; plot(real(eigp2),imag(eigp2),'xk');
```

Analisi spettrale

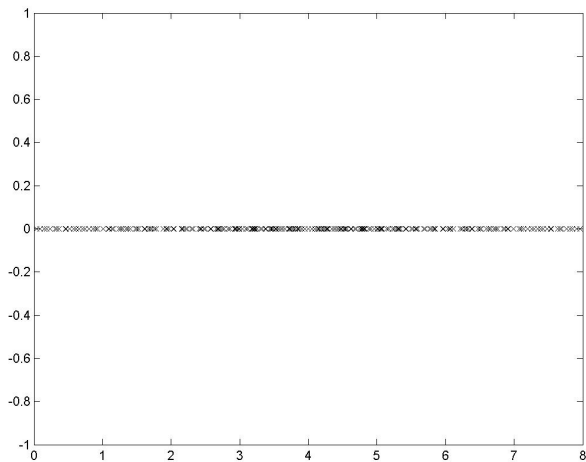


Figura: Autovalori del sistema originario

Analisi spettrale

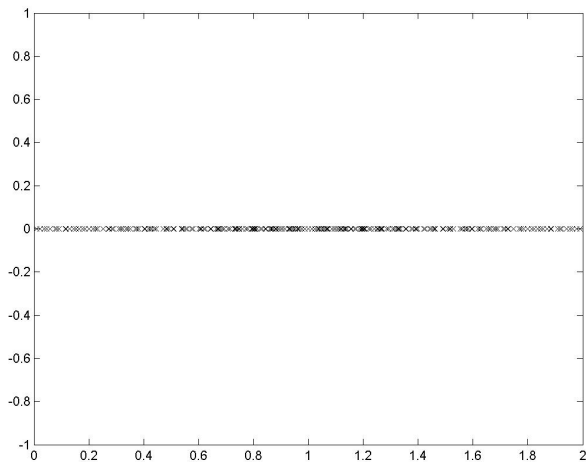


Figura: Autovalori del sistema preconditionato - diagonale

Analisi spettrale

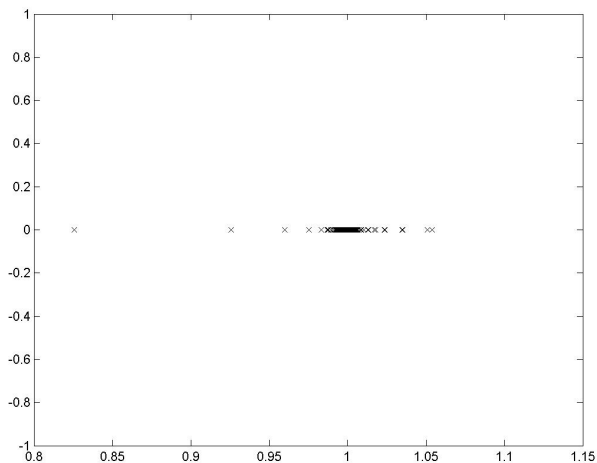


Figura: Autovalori del sistema preconditionato - Cholesky incompleto

FINE