TEO: $A \in \mathbb{R}^{n \times n}$ a <u>predom diag forte per righe</u>.
Allora il m. it di Jacobi è convergente

<u>Es</u>: $A = \begin{bmatrix} 3 & 0 & 0 & 1 \\ 1 & 3 & 0 & 1 \\ 1 & 0 & 3 & 1 \\ 1 & 0 & 0 & 3 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

- $H_J = -\frac{1}{3} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$, $c_J = \frac{1}{3} b$

A è PDF... !

- calcolo raggio spettr di H_J (in base al Teo, sarà <u>certam</u> < 1):

$$p(\lambda) = \det(H_J - \lambda I) = (-\lambda)^2 \left[(-\lambda)^2 - \frac{1}{9} \right] =$$

$$= (-\lambda)^2 \left(-\lambda + \frac{1}{3}\right)\left(-\lambda - \frac{1}{3}\right)$$

$\Rightarrow$ SPETTRO $\sigma(H_J) = \left\{ 0, \frac{1}{3}, -\frac{1}{3} \right\}$

$\Rightarrow$ RAGGIO SPETTRALE $\boxed{\rho(H_J) = \frac{1}{3}}$.

($x_k(j)$ = j-esima componente di $x_k \in \mathbb{R}^N$)

- $$x_{k+1} = \begin{bmatrix} -\frac{1}{3} x_k(N) + c_J(1) \\ -\frac{x_k(1) + x_k(N)}{3} + c_J(2) \\ \vdots \\ -\frac{x_k(1) + x_k(N)}{3} + c_J(N-1) \\ -\frac{1}{3} x_k(1) + c_J(N) \end{bmatrix}$$

$k = 0, 1, 2, \dots$

• posto x^* = la soluz di $Ax=b$; $e_k = x_k - x^*$;

$$\boxed{e_{k+1} = H_J\, e_k}$$

(1) $e_1 = H_J\, e_0 = -\frac{1}{3}\,\square\, e_0 = -\frac{1}{3}\begin{bmatrix} e_{04} \\ e_{01}+e_{04} \\ e_{01}+e_{04} \\ e_{01} \end{bmatrix}$

(2) $e_2 = H_J\, e_1 = -\frac{1}{3}\,\square\, e_1 = \frac{1}{9}\begin{bmatrix} e_{01} \\ e_{01}+e_{04} \\ e_{01}+e_{04} \\ e_{04} \end{bmatrix}$

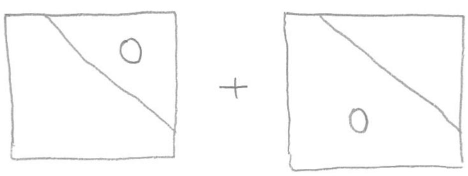
$\vdots$

$$\|e_k\| = \frac{1}{3^k}\left\|\begin{bmatrix} e_{04} \\ e_{01}+e_{04} \\ e_{01}+e_{04} \\ e_{01} \end{bmatrix}\right\| \xrightarrow[k\to\infty]{} 0$$

come $[\rho(H_J)]^k$

Metodo di GAUSS-SEIDEL

• $A \in \mathbb{R}^{n\times n}$ < invertibili, $\boxed{a_{kk} \neq 0}$, $b \in \mathbb{R}^n$

• $A =$ 

P $\qquad$ $A - P$ ← la parte strettam tr sup di A

↑ la parte tr inf di A

$$Ax = b \sim Px + (A-P)x = b$$

$$\sim Px = (P-A)x + b$$

$$\sim x = \underbrace{P^{-1}(P-A)}x + P^{-1}b \longrightarrow c_{GS}$$

$$= I - P^{-1}A = H_{GS}$$

è il m. it def da H_{GS} e c_{GS}.

Es: $A = \begin{bmatrix} 3 & 0 & 0 & 1 \\ 1 & 3 & 0 & 1 \\ 1 & 0 & 3 & 1 \\ 1 & 0 & 0 & 3 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

- $P = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}$, $P - A = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

- $P^{-1} = \begin{bmatrix} 1/3 & 0 & 0 & 0 \\ -1/9 & 1/3 & 0 & 0 \\ -1/9 & 0 & 1/3 & 0 \\ -1/9 & 0 & 0 & 1/3 \end{bmatrix}$; $H_{GS} = \begin{bmatrix} 0 & 0 & 0 & -1/3 \\ 0 & 0 & 0 & -2/9 \\ 0 & 0 & 0 & -2/9 \\ 0 & 0 & 0 & 1/9 \end{bmatrix}$

$$\Rightarrow \sigma(H_{GS}) = \left\{0, \frac{1}{9}\right\} \Rightarrow \boxed{\rho(H_{GS}) = \frac{1}{9}}$$

- $\rho(H_{GS}) = (\rho(H_J))^2$ (ma non è sempre così!...)

Oss: In generale, $P^{-1}(P-A)$ non è sparsa

e $\boxed{\text{costo } P^{-1}(P-A)\,x_k}$ > $\boxed{\text{costo soluz } P x_{k+1} = (P-A)\,x_k + b}$

$\sim 2n^2$ $\sim n^2$

$\Rightarrow$ ad ogni iterazione è più economico risolvere il sist (con matr tr inf) che fare il prodotto...

TEO: $A \in \mathbb{R}^{n \times n}$

- a predom diag forte per righe

oppure

- simmetrica def positiva

Allora il m. it di Gauss-Seidel è converg

```
//
clear;
//
Percorso = "/home/maurizio/Documents/Scienza/Scilab/Sistemi Lineari";
exec(Percorso + "/Metodi/EGPP.sci");
exec(Percorso + "/Metodi/SA.sci");
exec(Percorso + "/Metodi/SI.sci");
//
T_J = 0;
T_EG = 0;
NE = [];
//
N = 5000;
           //  100:   TJ = 1.2d-1   TG = 2.3d-1   EJ = 1.4d-14   EG = 3.29d-17
           //  500:   TJ = 1.8d-1   TG = 8.2
           // 1000:   TJ = 2.8d-1   TG = 62
           // 2000:   TJ = 5.4d-1   troppa memoria necessaria per EGPP
           // 3000:   TJ = 8.0d-1   troppa memoria necessaria per EGPP
           // 4000:   troppa memoria per costruire A; senza costruire A:
           //         TJ = 1.0
           // 5000:   TJ = 1.2
//
A = 3*eye(N,N); A(2:N,1) = ones(N-1,1); A(1:N-1,N) = ones(N-1,1);
b = ones(N,1);
//
c = b/3;
//
x = zeros(N,1);
xn = x;
x_esatta = ones(N,1)/6; x_esatta(1) = 1/4; x_esatta(N) = 1/4;
tic();
for k = 1:30,
    s = -(x(1)+x(N))/3;
    for i = 2:N-1, xn(i) = s + c(i); end;
    xn(1) = c(1) - x(N)/3;
    xn(N) = c(N) - x(1)/3;
    x = xn;
    NE(k) = norm(x-x_esatta);
end;
T_J = toc();
//
for k = find(NE == 0), NE(k) = %eps; end;
printf("\nerrore relativo J = %4.3e\n",norm(x - x_esatta)/norm(x_esatta));
RES = zeros(N,1);
RES(1) = 3*x(1) + x(N) - b(1);
for i = 2:N-1, RES(i) = 3*x(i) + x(1) + x(N) - b(i); end;
RES(N) = x(1) + 3*x(N) - b(N);
printf("\nmisura residuo = %4.3e\n",norm(RES)/norm(b));
printf("\ntempo per Jacobi (N = %d) = %4.3e\n",N, T_J);
//
scf(0);clf();plot(1:30,log10(NE));xgrid();
//
tic();
[S,D,P,info] = EGPP(A);
[c,info] = SA(S,P*b,"disabilita");
[x_EG, info] = SI(D,c,"disabilita");
T_EG = toc();
//
printf("\nerrore relativo EG = %4.3e\n",norm(x_EG - x_esatta)/norm(x_esatta));
printf("\ntempo per Gauss (N = %d)= %4.3e\n", N, T_EG);
//
```