

Metodo di JACOBI

- $A \in \mathbb{R}^{n \times n}$  { invertibili  
con  $a_{kk} \neq 0$  ,  $b \in \mathbb{R}^n$

•  $A = \text{diag}(a_{11}, \dots, a_{nn}) + M$

$Ax = b \sim \text{diag}(a_{11}, \dots, a_{nn})x = -Mx + b$

$\sim x = \underbrace{-\text{diag}(a_{11}, \dots, a_{nn})^{-1}}_{H_j} Mx +$

$\underbrace{+\text{diag}(a_{11}, \dots, a_{nn})^{-1} b}_{c_j}$

è il m. it def da  $H_j$  e  $c_j$ .

Es.  $A = \begin{bmatrix} 3 & -1 & 0 \\ 0 & 1 & 4 \\ 1 & 6 & 7 \end{bmatrix}$  ,  $b = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$

Oss:  $(a_{kk} \neq 0)$  ...  $\begin{bmatrix} 3 & & \\ & 1 & \\ & & 7 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 4 \\ 1 & 6 & 0 \end{bmatrix} \xrightarrow{M}$

$\Rightarrow H_J = - \begin{bmatrix} 1/3 & & \\ & 1 & \\ & & 1/7 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 4 \\ 1 & 6 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1/3 & 0 \\ 0 & 0 & -4 \\ -1/7 & -6/7 & 0 \end{bmatrix}$  ,  $c_J = \begin{bmatrix} 1/3 \\ 0 \\ 3/7 \end{bmatrix}$

TEO:  $A \in \mathbb{R}^{n \times n}$  a predom di'op forte per righe.

Allora il m. it di Jacobi è convergente

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

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

$A$  è PDF...!

• calcolo raggio spettro di  $H_J$  (in base al Teo, farà certam  $< 1$ ):

$$p(\lambda) = \det(H_J - \lambda I) = (-\lambda)^2 \left[ (-\lambda)^2 - \frac{1}{9} \right] =$$
$$= (-\lambda)^2 \left( -\lambda + \frac{1}{3} \right) \left( -\lambda - \frac{1}{3} \right)$$

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

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

•  $x_{k+1} = \begin{bmatrix} -\frac{1}{3} x_k(N) + c_J(1) \\ = \frac{x_k(1) + x_k(N)}{3} + c_J(2) \\ \vdots \\ -\frac{x_k(1) + x_k(N)}{3} + c_J(N-1) \\ -\frac{1}{3} x_k(1) + c_J(N) \end{bmatrix}$   $\dots k = 0, 1, 2, \dots$

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

$$e_{k+1} = H_J e_k$$

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

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

⋮

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

come  $[p(H_J)]^k$

## Metodo di GAUSS-SEIDEL

- $A \in \mathbb{R}^{n \times n}$   $\left\{ \begin{array}{l} \text{invertibili} \\ a_{kk} \neq 0 \end{array} \right.$ ,  $b \in \mathbb{R}^n$

- $A = \begin{bmatrix} \diagdown & & & \\ & \circ & & \\ & & \diagdown & \\ & & & \circ \end{bmatrix} + \begin{bmatrix} \diagdown & & & \\ & \circ & & \\ & & \diagdown & \\ & & & \circ \end{bmatrix}$   
 $\begin{matrix} P & & A-P \end{matrix}$   
 la parte tr inf di  $A$   $\leftarrow$  la parte strettam tr sup di  $A$

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

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

$$\sim x = \underbrace{P^{-1}(P-A)}_{= I - P^{-1}A} x + \underbrace{P^{-1}b}_{\text{CGS}} \rightarrow \text{CGS}$$
$$= I - P^{-1}A = H_{GS}$$

è il m.it def da  $H_{GS}$  e  $C_{GS}$ .

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

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

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

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

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

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

$$e \begin{array}{|l} \text{costo } P^{-1}(P-A) x_k \\ \hline \end{array} > \begin{array}{|l} \text{costo soluz} \\ P x_{k+1} = (P-A) x_k + b \\ \hline \end{array}$$

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

$\Rightarrow$  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

```

0001 //
0002 clear;
0003 //
0004 Percorso = "/home/maurizio/Documents/Scienza/Scilab/Sistemi Lineari";
0005 exec(Percorso + "/Metodi/EGPP.sci");
0006 exec(Percorso + "/Metodi/SA.sci");
0007 exec(Percorso + "/Metodi/SI.sci");
0008 //
0009 T_J = 0;
0010 T_EG = 0;
0011 NE = [];
0012 //
0013 N = 5000;
0014 // 100: TJ = 1.2d-1 TG = 2.3d-1 EJ = 1.4d-14 EG = 3.29d-17
0015 // 500: TJ = 1.8d-1 TG = 8.2
0016 // 1000: TJ = 2.8d-1 TG = 62
0017 // 2000: TJ = 5.4d-1 troppa memoria necessaria per EGPP
0018 // 3000: TJ = 8.0d-1 troppa memoria necessaria per EGPP
0019 // 4000: troppa memoria per costruire A; senza costruire A:
0020 // TJ = 1.0
0021 // 5000: TJ = 1.2
0022 //
0023 A = 3*eye(N,N); A(2:N,1) = ones(N-1,1); A(1:N-1,N) = ones(N-1,1);
0024 b = ones(N,1);
0025 //
0026 c = b/3;
0027 //
0028 x = zeros(N,1);
0029 xn = x;
0030 x_esatta = ones(N,1)/6; x_esatta(1) = 1/4; x_esatta(N) = 1/4;
0031 tic();
0032 for k = 1:30,
0033     s = -(x(1)+x(N))/3;
0034     for i = 2:N-1, xn(i) = s + c(i); end;
0035     xn(1) = c(1) - x(N)/3;
0036     xn(N) = c(N) - x(1)/3;
0037     x = xn;
0038     NE(k) = norm(x-x_esatta);
0039 end;
0040 T_J = toc();
0041 //
0042 for k = find(NE == 0), NE(k) = %eps; end;
0043 printf("\nerrore relativo J = %4.3e\n", norm(x - x_esatta)/norm(x_esatta));
0044 RES = zeros(N,1);
0045 RES(1) = 3*x(1) + x(N) - b(1);
0046 for i = 2:N-1, RES(i) = 3*x(i) + x(1) + x(N) - b(i); end;
0047 RES(N) = x(1) + 3*x(N) - b(N);
0048 printf("\nmisura residuo = %4.3e\n", norm(RES)/norm(b));
0049 printf("\ntempo per Jacobi (N = %d) = %4.3e\n", N, T_J);
0050 //
0051 scf(0);clf();plot(1:30,log10(NE));xgrid();
0052 //
0053 tic();
0054 [S,D,P,info] = EGPP(A);
0055 [c,info] = SA(S,P*b,"disabilita");
0056 [x_EG, info] = SI(D,c,"disabilita");
0057 T_EG = toc();
0058 //
0059 printf("\nerrore relativo EG = %4.3e\n", norm(x_EG - x_esatta)/norm(x_esatta));
0060 printf("\ntempo per Gauss (N = %d) = %4.3e\n", N, T_EG);
0061 //

```