① $(\hat{S}, \hat{D}, \hat{P}) = \widehat{EGP}(\hat{A})$

② $\hat{c} = \widehat{SA}(\hat{S}, \hat{P}\hat{b})$

③ $\hat{x} = \widehat{SI}(\hat{D}, \hat{c})$

<u>Es</u>: $A = \begin{pmatrix} \gamma & 1 \\ 1 & 0 \end{pmatrix}$, $\gamma \in (0,1)$, $b \in \mathbb{R}^2$

supponendo che

- $\hat{A} = A$, $\hat{b} = b$
- $\widehat{EGP}(A) = EGP(A)$
- $\widehat{SA}(S, Pb) = \hat{c} \neq c$ con $\dfrac{\|\hat{c} - c\|}{\|c\|} < u$
- $\widehat{SI}(D, \hat{c}) = SI(D, \hat{c})$

si ha:

x_* soluz di: $Dx = c$

$\hat{x}$ soluz di: $Dx = \hat{c} = c + f$, $\alpha = u$

<u>Teo condiz</u> $\Rightarrow \dfrac{\|\hat{x} - x_*\|}{\|x_*\|} \leq 2u \, c(D)$

<u>Ma</u>: $c(D) \geq \dfrac{1}{\gamma}\sqrt{1 + \dfrac{1}{\gamma^2}}$ e $\lim_{\gamma \to 0} c(D) = +\infty$

Si può rendere $c(D)$ arb <u>GRANDE</u> scegliendo γ opportunam <u>PICCOLO</u>.

Il procedim riduce la soluz di $Ax=b$ a quella di $Dx=c$, ma i due sistemi, benché EQUIVALENTI (hanno la stessa soluz) NON hanno le stesse propr di condizionam:

$c(A) \ll c(D)$!

Il procedim, SODDISFACENTE op in $\mathbb{R}$, è NON SODDISFACENTE op in $F(\beta, m)$!

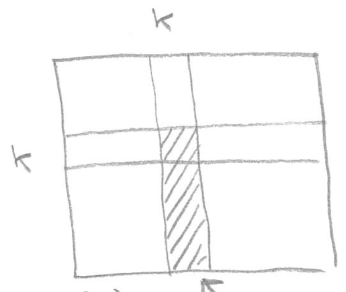
Rimedio: EGPP

(Elim di Gauss con Pivoting Parziale)

al passo k si utilizza come pivot l'elemento

$$\max_{i \geq k} |a_{ik}^{(k)}|$$

elem di massimo modulo in



$A^{(k)}$

$$\text{Es:}\quad \overset{A}{\begin{bmatrix}1&0&1\\2&1&-1\\1&2&1\end{bmatrix}} \xrightarrow{P_{12}} \begin{bmatrix}2&1&-1\\1&0&1\\1&2&1\end{bmatrix} \xrightarrow{H_1} \begin{bmatrix}2&1&-1\\0&-1/2&3/2\\0&3/2&3/2\end{bmatrix} \cdots$$

$$\cdots \xrightarrow{P_{32}} \begin{bmatrix}2&1&-1\\0&3/2&3/2\\0&-1/2&3/2\end{bmatrix} \xrightarrow{H_2} \begin{bmatrix}2&1&-1\\0&3/2&3/2\\0&0&2\end{bmatrix} = D$$

$$S = \begin{bmatrix}1&0&0\\1/2&1&0\\1/2&-1/3&1\end{bmatrix}, \quad P = P_{32}P_{12}; \quad \boxed{EGPP(A) = (S, D, P)}$$

Oss: utilizz EGPP:

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

$$\boxed{c(D) \leq F(n)\, c(A)}$$

dip solo dalla dim della matrice

in particolare NON si può avere crescita illimitata del n. di cond nel passaggio da A a D.

• uso della fatt QR in $F(\beta, m)$:

(1) $A = UT \Rightarrow \|A\| = \|UT\| = \max_{\|v\|=1} \|U(Tv)\|$

$= \max_{\|v\|=1} \|Tv\|$

$= \|T\|$

ortogonale $\Rightarrow \forall w$, $\|Uw\| = \|w\|$

(2) $T^{-1} = A^{-1}U \Rightarrow \|T^{-1}\| = \|A^{-1}U\| = \max_{\|v\|=1} \|A^{-1}(Uv)\|$

$= \max_{\|Uv\|=1} \|A^{-1}(Uv)\| = \|A^{-1}\|$

Q.d.: $\boxed{c(T) = \|T\| \, \|T^{-1}\| = c(A)}$.

<u>In pratica</u>:

utilizz EGPP o qr:

$\hat{x}$ t.c. $(A+E)\hat{x} = b$ con $\|E\| \leq F(n)u\|A\|$

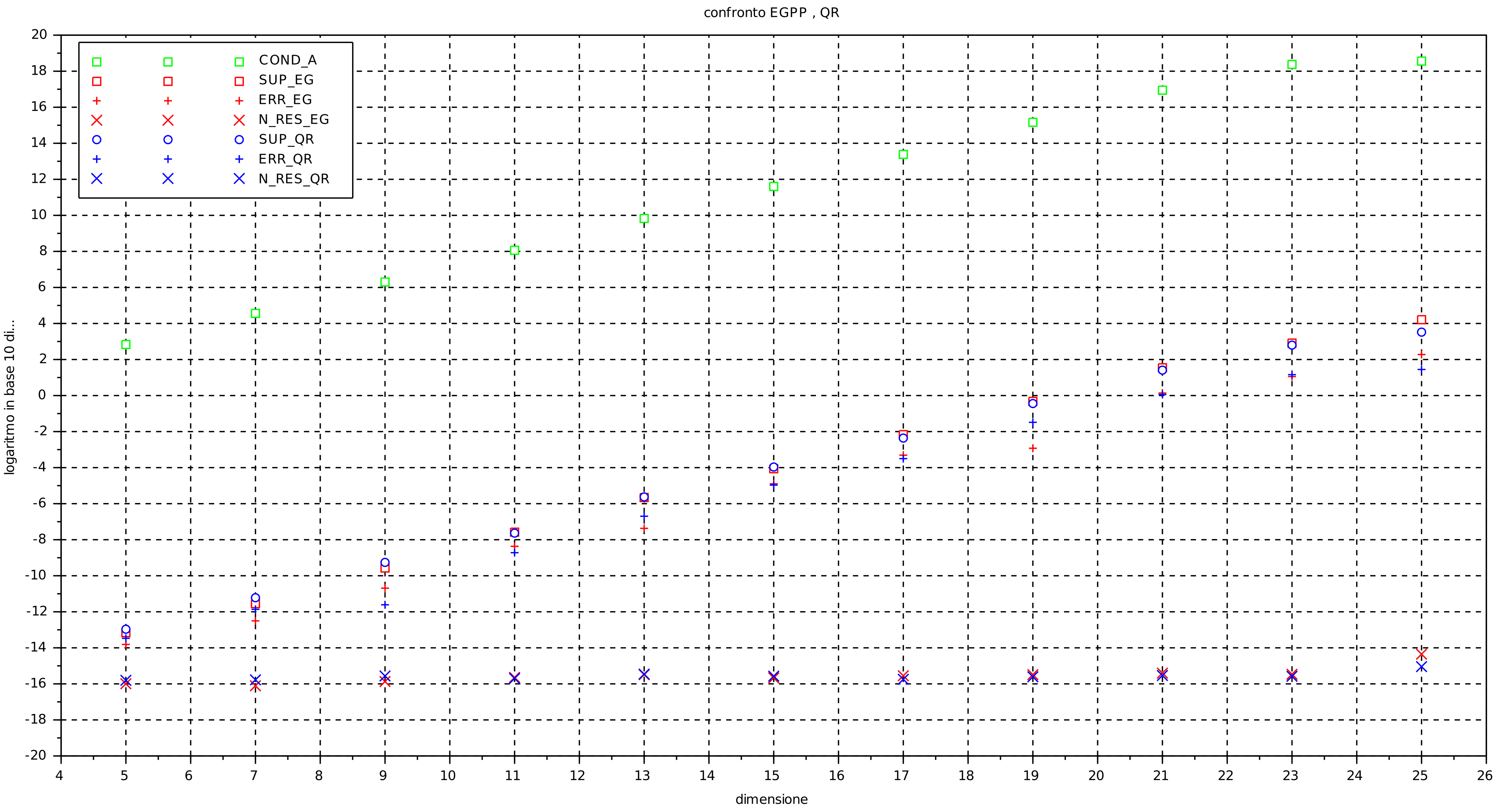
- $F(n) \approx n$ quasi sempre con EGPP, ma per qualche A si ha $F(n) \approx 2^n$;

- $F(n) \approx n$ utilizz QR calcolata opportunam (NON con GS!): l'istruzione qr di Scilab va bene.

Ancora: Teo condiz ...

```
//
// Esempio con matrice di Vandermonde. La matrice ha un numero di
// condizionamento rapidamente crescente con la dimensione del sistema.
//
// Il sistema è risolto in due diversi modi: utilizzando EGPP ed utilizzando
// qr.
//
// Osservazione: la misura relativa dei residui N_RES (paragonabile nei due casi)
// indica che il sistema è stato risolto "bene" dal calcolatore. L'errore
// relativo, rapidamente crescente con la dimensione, è dovuto al pessimo
// condizionamento del sistema.
//
clear;
//
Percorso = "PERCORSO files seguenti";  // <------ *** MODIFICARE ***
exec(Percorso + "EGPP.sci");
exec(Percorso + "SA.sci");
exec(Percorso + "SI.sci");
//
// variabili utili
//
DIM = [];
COND_A = [];
ERR_EG = [];
N_RES_EG = [];
ERR_QR = [];
N_RES_QR = [];
SUP_EG = [];
SUP_QR = [];
//
// ciclo sulla dimensione del sistema
//
for N = 5:2:25,
DIM($+1) = N;
//
// generazione della matrice
//
colonna = linspace(0,1,N)';
A = zeros(N,N);
for k=1:N,
    A(:,k) = colonna.^(k-1);
end;
//
// soluzione esatta e termine noto corrispondente
//
x_esatta = ones(N,1)/sqrt(2);
b = A*x_esatta;
//
// numero di condizionamento della matrice
//
COND_A($+1) = cond(A);
//
// soluzione del sistema con EGPP, SA e SI
//
[S,D,P,info_EGPP] = EGPP(A);
[c_EG,info_SA] = SA(S,P*b,"disabilita");
[x_EG,info_SI] = SI(D,c_EG,"disabilita");
//
// soluzione con qr, moltiplicazione e SI
//
[U,T] = qr(A);
c_QR = U' * b;
[x_QR,info_SI] = SI(T,c_QR,"disabilita");
//
// misura relativa degli errori
//
ERR_EG($+1) = norm(x_EG - x_esatta)/norm(x_esatta);
ERR_QR($+1) = norm(x_QR - x_esatta)/norm(x_esatta);
//
// misura relativa dei residui
```

```
//
N_RES_EG($+1) = norm(A*x_EG - b)/norm(b);
N_RES_QR($+1) = norm(A*x_QR - b)/norm(b);
//
// uso del Teorema di condizionamento
//
SUP_EG($+1) = COND_A($)*N_RES_EG($);
SUP_QR($+1) = COND_A($)*N_RES_QR($);
end;
//
// grafici
//
scf(0);clf(0);
plot(DIM,log10(COND_A),"gs",...
     DIM,log10(SUP_EG),"rs",DIM,log10(ERR_EG),"r+",DIM,log10(N_RES_EG),"rx",...
     DIM,log10(SUP_QR),"bo",DIM,log10(ERR_QR),"b+",DIM,log10(N_RES_QR),"bx");
xlabel("dimensione");
ylabel("logaritmo in base 10 di...");
legend("COND_A","SUP_EG","ERR_EG","N_RES_EG","SUP_QR","ERR_QR","N_RES_QR",2);
title("confronto EGPP , qr");
```



confronto EGPP , QR
COND_A
SUP_EG
ERR_EG
N_RES_EG
SUP_QR
ERR_QR
N_RES_QR
logaritmo in base 10 di...
dimensione
20
18
16
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26