

Un algoritmo rank-revealing per la risoluzione nel senso dei minimi quadrati di sistemi lineari a rango non pieno

G. Rodriguez

Dipartimento di Matematica e Informatica, Università di Cagliari
viale Merello 92, 09123 Cagliari, Italy

in collab. con A. Aricò (Dip. di Matematica e Informatica, Cagliari)

Giornate di Algebra Lineare Numerica ed Applicazioni
Padova, 26–27 febbraio 2007

Minimi quadrati mediante eliminazione Gaussiana

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{m \times n}, \quad \text{rank}(A) = n \leq m$$

- Metodo di Peters-Wilkinson [1970]

$$PAQ = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} U, \quad L_1, U \in \mathbb{R}^{n \times n}$$

$$L^T L \mathbf{y} = L^T (P\mathbf{b}), \quad U(Q^T \mathbf{x}) = \mathbf{y}$$

Applicato a matrici sparse in [Björck et al. 1988].

Minimi quadrati mediante eliminazione Gaussiana

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{m \times n}, \quad \text{rank}(A) = n \leq m$$

- Metodo di Peters-Wilkinson [1970]

- Metodo della matrice aumentata

[Siegel 1965, Bartel et al. 1970, Björck 1992]

$$\begin{bmatrix} I & A \\ A^* & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}, \quad \mathbf{r} = \mathbf{b} - A\mathbf{x}$$

Applicato con successo in [Arioli et al. 1989] e
esteso in [Duff et al. 1990] al caso di matrici sparse con
pivoting di Bunch-Kaufman.

Minimi quadrati e complemento di Schur

Metodo proposto per matrici di Toeplitz in [Kailath et al. 1994] e studiato e implementato in [GR 2006] per matrici con struttura di Toeplitz e Cauchy generalizzata.

$$A\mathbf{x} = \mathbf{b} \quad \longrightarrow \quad M_A = \left[\begin{array}{cc|c} I_m & A & 0 \\ A^* & 0 & A^*\mathbf{b} \\ \hline 0 & I_n & 0 \end{array} \right]$$

$$\mathcal{S}_{m+n}(M_A) = (A^*A)^{-1}A^*\mathbf{b} = \mathbf{x}_{LS}$$

Poiché M_A ha struttura di *displacement* è possibile applicare l'**algoritmo di Schur generalizzato**.

Inoltre **la matrice pseudoinversa ha struttura di *displacement***

$$\mathbf{b} = I_m \quad \Rightarrow \quad \mathcal{S}_{m+n}(M_A) = A^\dagger,$$

Calcolo del complemento di Schur di M_A

Schema di calcolo

- la struttura viene convertita da Toeplitz a **Cauchy generalizzata**, mediante trasformate veloci
- viene applicato l'**algoritmo di Schur generalizzato**

Calcolo del complemento di Schur di M_A

Schema di calcolo

- la struttura viene convertita da Toeplitz a **Cauchy generalizzata**, mediante trasformate veloci
- viene applicato l'**algoritmo di Schur generalizzato**

Vantaggi

- M_A , pur avendo struttura di *displacement*, non è Toeplitz-like, mentre M_C è **Cauchy-like**
- il passaggio alla struttura di Cauchy consente l'uso del **pivoting parziale** o totale
- **algoritmo fast** $O(27\gamma(m+n)^2)$ e **bassa occupazione di memoria** $O(3\gamma(m+n))$ ($\gamma = \text{rank}_\Delta(M_A)$)
- facilmente applicabile ad altre strutture di *displacement*

Restrizione del pivoting

Nel calcolo del complemento di Schur $S_r(M)$, l'azione del pivoting parziale (o totale) deve essere **ristretta alle prime r righe** (e colonne) di M . Infatti se

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \quad P = \begin{bmatrix} P_r & 0 \\ 0 & I_{m-r} \end{bmatrix} \quad \text{e} \quad Q = \begin{bmatrix} Q_r & 0 \\ 0 & I_{n-r} \end{bmatrix}$$

allora

$$PMQ = \begin{bmatrix} P_r M_{11} Q_r & P_r M_{12} \\ M_{21} Q_r & M_{22} \end{bmatrix}$$

e

$$S_r(PMQ) = S_r(M).$$

Blocchi parzialmente ricostruibili

$$C_{ij} = \frac{\phi_i^* \cdot \psi_j}{t_i - s_j}$$

$$D_t C - C D_s = G_C H_C^*$$

Blocchi parzialmente ricostruibili

$$C_{ij} = \frac{\phi_i^* \cdot \psi_j}{t_i - s_j}$$

$$D_{\mathbf{t}}C - CD_{\mathbf{s}} = G_C H_C^* \quad \Rightarrow \quad \mathcal{D}_L M_C - M_C \mathcal{D}_R = G_{M_C} H_{M_C}^*$$

$$M_C = \begin{bmatrix} I_m & C & 0 \\ C^* & 0 & C^* \\ 0 & I_n & 0 \end{bmatrix}, \quad \begin{aligned} \mathcal{D}_L &= D_{\mathbf{t}} \oplus D_{\mathbf{s}} \oplus D_{\mathbf{s}}, \\ \mathcal{D}_R &= D_{\mathbf{t}} \oplus D_{\mathbf{s}} \oplus D_{\mathbf{t}}, \end{aligned}$$

Blocchi parzialmente ricostruibili

$$C_{ij} = \frac{\phi_i^* \cdot \psi_j}{t_i - s_j}$$

$$D_{\mathbf{t}}C - CD_{\mathbf{s}} = G_C H_C^* \quad \Rightarrow \quad \mathcal{D}_L M_C - M_C \mathcal{D}_R = G_{M_C} H_{M_C}^*$$

$$M_C = \begin{bmatrix} I_m & C & 0 \\ C^* & 0 & C^* \\ 0 & I_n & 0 \end{bmatrix}, \quad \begin{aligned} \mathcal{D}_L &= D_{\mathbf{t}} \oplus D_{\mathbf{s}} \oplus D_{\mathbf{s}}, \\ \mathcal{D}_R &= D_{\mathbf{t}} \oplus D_{\mathbf{s}} \oplus D_{\mathbf{t}}, \end{aligned}$$

Alcuni blocchi sono **parzialmente ricostruibili**:

$$\text{blocco } (3, 2) \longrightarrow D_s I_n - I_n D_s = 0$$

È necessario memorizzare a parte gli elementi non ricostruibili e tener conto del pivoting.

Ottimizzazione dei parametri ξ e η

$$Z_{\xi,m}A - AZ_{\eta,n} = G_A H_A^*, \quad \xi, \eta \in \mathbb{C} \setminus \{0\}$$

$$Z_{\phi,k} = \begin{pmatrix} 0 & 0 & \cdots & 0 & \phi \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \in \mathbb{C}^{k \times k}.$$

Ottimizzazione dei parametri ξ e η

$$Z_{\xi,m}A - AZ_{\eta,n} = G_A H_A^*, \quad \xi, \eta \in \mathbb{C} \setminus \{0\}$$

$$D_t C - C D_s = G_C H_C^*$$

$$C_{ij} = \frac{\phi_i^* \cdot \psi_j}{t_i - s_j}, \quad t_i^m = \xi, \quad s_j^n = \eta$$

Ottimizzazione dei parametri ξ e η

$$Z_{\xi,m}A - AZ_{\eta,n} = G_A H_A^*, \quad \xi, \eta \in \mathbb{C} \setminus \{0\}$$

$$C_{ij} = \frac{\phi_i^* \cdot \psi_j}{t_i - s_j}, \quad t_i^m = \xi, \quad s_j^n = \eta$$

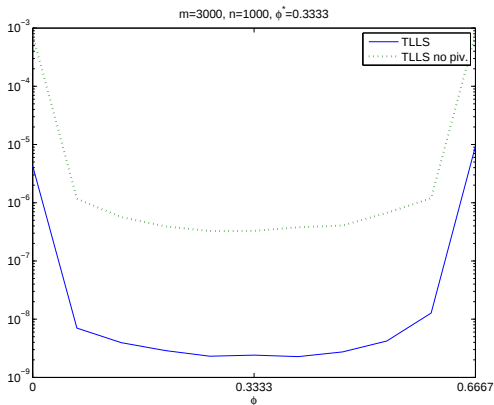
Posto $\xi = 1$ e $\eta = e^{i\pi\varphi}$, sia

$$\varphi^* = \arg \max_{\varphi} \min_{i,j} |t_i - s_j|.$$

Allora

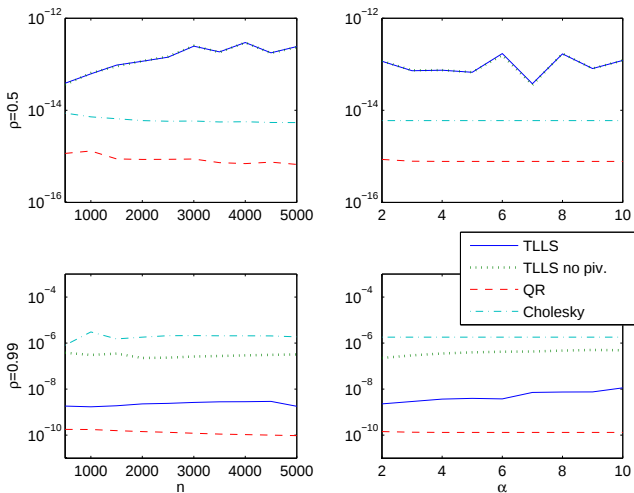
$$\varphi^* = \frac{\gcd(m, n)}{m} = \frac{g}{m}, \quad \min_{i,j} |t_i - s_j| = 2 \sin \frac{\pi g}{2mn}.$$

Ottimizzazione dei parametri ξ e η



Matrice KMS ($a_{ij} = \rho^{|i-j|}$)
con $(m, n) = (3000, 1000)$, $\rho = .99$, $\kappa = 3.7 \cdot 10^4$

Risultati numerici



Matrice KMS, $m = 2n$ (sin.), $\alpha = \frac{m}{n}$ e $n = 2000$ (destra)

Dipendenza dal condizionamento

Condizionamento di $\mathbf{x}_{LS}$ rispetto ad una perturbazione in A

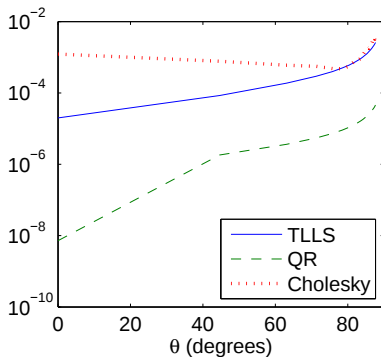
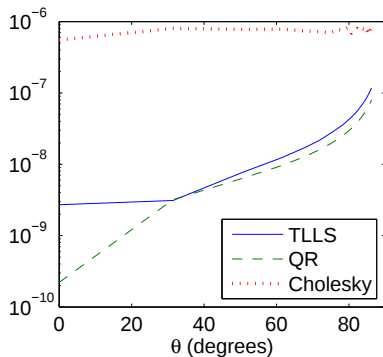
$$\kappa_{LS} = \kappa(A) + \frac{\kappa(A)^2 \tan \theta}{\eta}$$

con $\kappa(A) = \|A\| \cdot \|A^\dagger\|$, $\eta = \frac{\|A\| \cdot \|\mathbf{x}\|}{\|A\mathbf{x}\|} \in [1, \kappa(A)]$, and

$$\theta = \arccos \frac{\|A\mathbf{x}\|}{\|\mathbf{b}\|} = \angle(\mathbf{b}, A\mathbf{x}) \in \left[0, \frac{\pi}{2}\right]$$

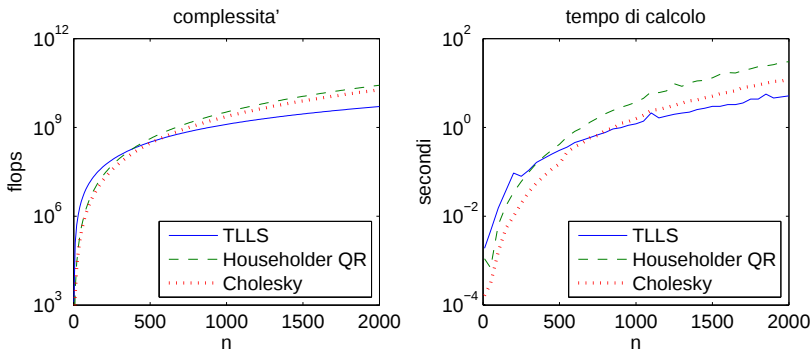
- Householder QR è influenzato da κ_{LS}
- Eq. normali + Cholesky dipende da $\kappa(A)^2$ e non da η, θ
- TLLS ?

Dipendenza dal condizionamento



Matrice KMS, $(m, n) = (1500, 500)$, $\rho = .99, .999$
 $\kappa(A) = 3.7 \cdot 10^4, 1.1 \cdot 10^6, \eta \simeq 1$

Complessità computazionale - metodi non strutturati

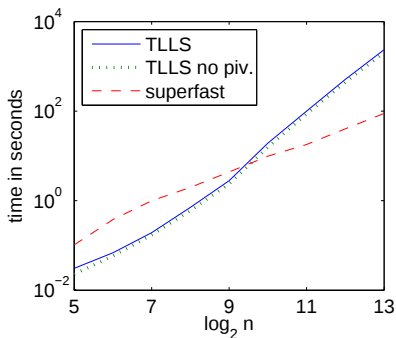
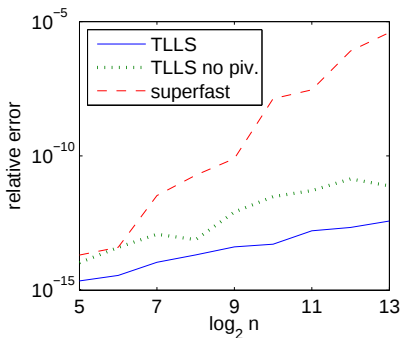


Confronto tra qr e cho1 di Matlab e la versione MEX di TLLS

AMD Athlon 64 3200+, GNU/Linux x86_64, Matlab 64bit

Matrice KMS ($\rho = .99$), $m = 2n$

Complessità computazionale - metodo superfast



Confronto tra le versioni Matlab di TLLS e di un algoritmo superfast
[Van Barel, Heinig, Kravanja 2003]
Matrice rand-Toeplitz, $m = 2n$

Complessità: *crossover* risp. ai metodi non strutturati

La dipendenza da m è **lineare** per i metodi non strutturati e **quadratica** per TLLS.

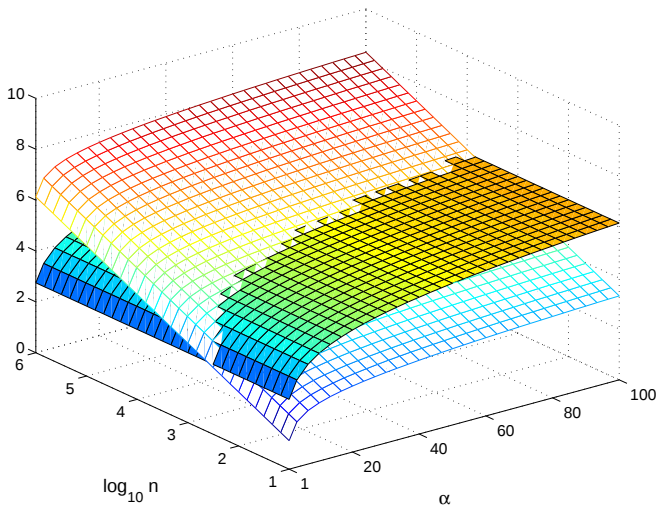
$$\text{Cholesky} \longrightarrow n^2(m + \frac{1}{3}n)$$

$$\text{Householder QR} \longrightarrow 2n^2(m - \frac{1}{3}n)$$

$$\text{TLLS} \longrightarrow 143m^2 + 284mn + 133n^2$$

Questo significa che per ogni n , all'aumentare di m , si arriva ad un punto in cui i metodi non strutturati risultano più efficienti.

Complessità: *crossover* risp. ai metodi non strutturati



$\log(\text{Complessità}/n^2)$ per Householder QR (grid) e TLLS (surf)

Sistema lineare a rango non pieno

- Eliminiamo l'ipotesi che la matrice A sia sovradeterminata e a rango pieno

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{m \times n}, \quad \begin{cases} m \leq n \\ \text{rank}(A) \leq \min(m, n) \end{cases}$$

- Applichiamo l'algoritmo TLSS

$$M_A = \left[\begin{array}{cc|c} I_m & A & 0 \\ A^* & 0 & A^*\mathbf{b} \\ \hline 0 & I_n & 0 \end{array} \right] \longrightarrow \mathcal{S}_{m+n}(M_A) = ?$$

Nel caso a rango pieno si ha

$$\mathbf{x}_{LS} = \mathcal{S}_{m+n}(M_A) = (A^*A)^{-1}A^*\mathbf{b}$$

Fase 1

Applichiamo l'algoritmo di eliminazione di Gauss alle prime m colonne. Il **pivoting** non è indispensabile per il funzionamento dell'algoritmo (lo è per la sua stabilità).

$$\left[\begin{array}{c|cc} I_m & A & 0 \\ \hline A^* & 0 & A^* \mathbf{b} \\ 0 & I_n & 0 \end{array} \right] \longrightarrow \left[\begin{array}{c|cc} I_m & A & 0 \\ \hline 0 & -A^* A & A^* \mathbf{b} \\ 0 & I_n & 0 \end{array} \right]$$

Questa fase è **equivalente** alla costruzione delle equazioni normali.

$A^* A$ in generale è singolare

Fase 2

Supponiamo inizialmente che A^*A sia *ben ordinata*:

$$-A^*A = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

con $\text{rank}(A) = k$ e G_{11} quadrata $k \times k$ non singolare.

Applichiamo k passi dell'algoritmo di Gauss alla sottomatrice

$$\left[\begin{array}{c|cc} I_m & A & 0 \\ \hline 0 & -A^*A & A^*\mathbf{b} \\ 0 & I_n & 0 \end{array} \right] \longrightarrow \left[\begin{array}{cc} -A^*A & A^*\mathbf{b} \\ I_n & 0 \end{array} \right]$$

Fase 2

Supponiamo inizialmente che A^*A sia *ben ordinata*:

$$-A^*A = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

con $\text{rank}(A) = k$ e G_{11} quadrata $k \times k$ non singolare.

Applichiamo k passi dell'algoritmo di Gauss alla sottomatrice

$$\left[\begin{array}{c|c} -A^*A & A^*\mathbf{b} \\ \hline I_n & 0 \end{array} \right] = \left[\begin{array}{cc|c} G_{11} & G_{12} & \mathbf{c}_1 \\ G_{21} & G_{22} & \mathbf{c}_1 \\ \hline I_k & 0 & 0 \\ 0 & I_{n-k} & 0 \end{array} \right]$$

Fase 2

Supponiamo inizialmente che A^*A sia *ben ordinata*:

$$-A^*A = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

con $\text{rank}(A) = k$ e G_{11} quadrata $k \times k$ non singolare.

Applichiamo k passi dell'algoritmo di Gauss alla sottomatrice

$$\left[\begin{array}{cc|c} G_{11} & G_{12} & \mathbf{c}_1 \\ G_{21} & G_{22} & \mathbf{c}_1 \\ \hline I_k & 0 & 0 \\ 0 & I_{n-k} & 0 \end{array} \right] \longrightarrow \left[\begin{array}{cc|c} U_k & Z & \tilde{\mathbf{c}}_1 \\ 0 & O_{n-k} & \tilde{\mathbf{c}}_2 \\ \hline 0 & Y_1 & \mathbf{x}_1 \\ 0 & Y_2 & \mathbf{x}_2 \end{array} \right]$$

Risultato dell'algoritmo

$$\left[\begin{array}{cc|c} U_k & Z & \tilde{\mathbf{c}}_1 \\ 0 & O_{n-k} & \tilde{\mathbf{c}}_2 \\ \hline 0 & Y_1 & \mathbf{x}_1 \\ 0 & Y_2 & \mathbf{x}_2 \end{array} \right]$$

- soluzione di base di $A\mathbf{x} = \mathbf{b}$

$$\mathbf{x}_B = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

Risultato dell'algoritmo

$$\left[\begin{array}{cc|c} U_k & Z & \tilde{\mathbf{c}}_1 \\ 0 & O_{n-k} & \tilde{\mathbf{c}}_2 \\ \hline 0 & \mathbf{Y}_1 & \mathbf{x}_1 \\ 0 & \mathbf{Y}_2 & \mathbf{x}_2 \end{array} \right]$$

- soluzione di base di $A\mathbf{x} = \mathbf{b}$

$$\mathbf{x}_B = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

- base per il nucleo di A

$$Y = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} = [\mathbf{y}_1 \quad \cdots \quad \mathbf{y}_{n-k}]$$

Risultato dell'algoritmo

$$\left[\begin{array}{cc|c} U_k & Z & \tilde{\mathbf{c}}_1 \\ 0 & O_{n-k} & \tilde{\mathbf{c}}_2 \\ \hline 0 & \mathbf{Y}_1 & \mathbf{x}_1 \\ 0 & \mathbf{Y}_2 & \mathbf{x}_2 \end{array} \right]$$

- soluzione di base di $A\mathbf{x} = \mathbf{b}$

$$\mathbf{x}_B = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} -G_{11}^{-1}\mathbf{c}_1 \\ 0 \end{bmatrix}$$

- base per il nucleo di A

$$Y = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} = [\mathbf{y}_1 \quad \cdots \quad \mathbf{y}_{n-k}] = \begin{bmatrix} -G_{11}^{-1}G_{12} \\ I_{n-k} \end{bmatrix}$$

Rank revealing?

Rango numerico

$$r_\epsilon = \min_{\|E\|_2 \leq \epsilon} \text{rank}(A + E) \quad \Leftrightarrow \quad \sigma_{r_\epsilon} > \epsilon \geq \sigma_{r_\epsilon+1}$$

L'algoritmo si arresta quando

$$|M_{rr}| < \tau$$

C'è relazione tra ϵ e τ ?

Rank revealing LU [Chan 1984], [Hwang et al. 1992].

Stime del condizionamento per matrici triangolari [Higham 1987].

Pivoting nella fase 2

Nella fase 2 il pivoting (su righe e colonne) è indispensabile

Supponiamo di applicare il **pivoting totale**

$$\begin{bmatrix} -A^*A & A^*\mathbf{b} \\ I_n & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} -PA^*AQ & PA^*\mathbf{b} \\ Q & 0 \end{bmatrix}$$

e poniamo

$$G = -PA^*AQ, \quad Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}, \quad \mathbf{c} = PA^*\mathbf{b}.$$

Allora

$$\left[\begin{array}{cc|c} G_{11} & G_{12} & \mathbf{c}_1 \\ G_{21} & G_{22} & \mathbf{c}_1 \\ \hline Q_1 & Q_2 & 0 \end{array} \right] \longrightarrow \left[\begin{array}{cc|c} U_k & L^{-1}G_{12} & L^{-1}\mathbf{c}_1 \\ 0 & O_{n-k} & \mathbf{c}_2 - G_{21}G_{11}^{-1}\mathbf{c}_1 \\ \hline 0 & Q_2 - Q_1G_{11}^{-1}G_{12} & -Q_1G_{11}^{-1}\mathbf{c}_1 \end{array} \right]$$

Pivoting nella fase 2

$$\left[\begin{array}{cc|c} U_k & L^{-1}G_{12} & L^{-1}\mathbf{c}_1 \\ 0 & O_{n-k} & \mathbf{c}_2 - G_{21}G_{11}^{-1}\mathbf{c}_1 \\ \hline 0 & Q_2 - Q_1G_{11}^{-1}G_{12} & -Q_1G_{11}^{-1}\mathbf{c}_1 \end{array} \right]$$

- soluzione di base di $A\mathbf{x} = \mathbf{b}$

$$\mathbf{x}_B = -Q_1G_{11}^{-1}\mathbf{c}_1 = Q \begin{bmatrix} -G_{11}^{-1}\mathbf{c}_1 \\ 0 \end{bmatrix},$$

Pivoting nella fase 2

$$\left[\begin{array}{cc|c} U_k & L^{-1}G_{12} & L^{-1}\mathbf{c}_1 \\ 0 & O_{n-k} & \mathbf{c}_2 - G_{21}G_{11}^{-1}\mathbf{c}_1 \\ \hline 0 & Q_2 - Q_1 G_{11}^{-1}G_{12} & -Q_1 G_{11}^{-1}\mathbf{c}_1 \end{array} \right]$$

- soluzione di base di $A\mathbf{x} = \mathbf{b}$

$$\mathbf{x}_B = -Q_1 G_{11}^{-1}\mathbf{c}_1 = Q \begin{bmatrix} -G_{11}^{-1}\mathbf{c}_1 \\ 0 \end{bmatrix},$$

- base per il nucleo di A

$$Y = Q_2 - Q_1 G_{11}^{-1}G_{12} = Q \begin{bmatrix} -G_{11}^{-1}G_{12} \\ I_{n-k} \end{bmatrix}$$

Pivoting nella fase 1

Nella fase 1 il pivoting non è indispensabile

$$M = \left[\begin{array}{c|cc} I_m & A & 0 \\ \hline A^* & 0 & A^* \mathbf{b} \\ 0 & I_n & 0 \end{array} \right]$$

Pivoting nella fase 1

Nella fase 1 il pivoting non è indispensabile,
ma è determinante per la stabilità dell'algoritmo.

$$M_{\gamma} = \left[\begin{array}{c|cc} \gamma I_m & A & 0 \\ \hline A^* & 0 & \frac{1}{\gamma} A^* \mathbf{b} \\ 0 & I_n & 0 \end{array} \right]$$

Il parametro di scala γ regola l'*entità* del pivoting.

Pivoting nella fase 1

Nella fase 1 il pivoting non è indispensabile,
ma è determinante per la stabilità dell'algoritmo.

$$M_\gamma = \left[\begin{array}{c|cc} \gamma I_m & A & 0 \\ \hline A^* & 0 & \frac{1}{\gamma} A^* \mathbf{b} \\ 0 & I_n & 0 \end{array} \right]$$

Il parametro di scala γ regola l'*entità* del pivoting.

Per il metodo della matrice aumentata con equilibratura, è stato dimostrato [Björck 1992] che una buona scelta è

$$\gamma = 2^{-1/2} \sigma_n(A)$$

in quanto minimizza $\kappa_2(M_\gamma)$, e quindi la stima in avanti dell'errore sulla soluzione.

Strategie di pivoting

- pivoting totale (costoso)

Strategie di pivoting

- pivoting totale (costoso)
- pivoting parziale + *“butta via le colonne piccole”*

Strategie di pivoting

- pivoting totale (costoso)
- pivoting parziale + *“butta via le colonne piccole”*
- pivoting totale simmetrico (fattorizzazione LDL^T)

Strategie di pivoting

- pivoting totale (costoso)
- pivoting parziale + *“butta via le colonne piccole”*
- pivoting totale simmetrico (fattorizzazione LDL^T)
- pivoting di Bunch-Kaufman

Fattorizzazione LDL^T

Preserva la *simmetria* ed è *connessa* a Cholesky e QR.

$$\begin{bmatrix} I_m & A & 0 \\ A^* & 0 & A^* \mathbf{b} \\ 0 & I_n & 0 \end{bmatrix} \xrightarrow{1} \begin{bmatrix} I_m & 0 & 0 \\ 0 & -A^* A & A^* \mathbf{b} \\ 0 & I_n & 0 \end{bmatrix} \xrightarrow{2} \begin{bmatrix} I_m & 0 & 0 \\ 0 & D & 0 \\ 0 & 0 & \mathbf{x}_{LS} \end{bmatrix}$$

Pivoting totale simmetrico

- se al passo k il massimo è in posizione (ℓ, ℓ)

$$\text{pivot} = M_{\ell\ell}$$

- se il massimo è in posizione (r, s)

$$\text{pivot} = \begin{pmatrix} M_{ss} & M_{sr} \\ M_{rs} & M_{rr} \end{pmatrix} \quad (\text{tile pivot})$$

Da fare...

- implementazione nel caso strutturato
- test numerici (matrici con e senza struttura)
- indagare sulle varie strategie di pivoting
- confronto con RRQR, UTV, [Chan 1984]
- soluzione normale, *up/down-dating*, *subset selection*...
- matrici sparse?

Fine