



$$PP^T = P_{ij}P_{kl}P_{rs}\dots P_{uv}P_{uv}\dots P_{rs}P_{kl}P_{ij} = I$$

$P$  è ortogonale.

# Fattorizzazione di una matrice qualunque con trasformazioni elementari di Gauss

Lo scopo è quello di risolvere sistemi in cui  $A$ , pur essendo non singolare, non è fattorizzabile nella forma  $LR$ .

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Ax = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Se si permutano le due equazioni si ottiene un sistema equivalente e fattorizzabile secondo Gauss. Ciò significa premoltiplicare ambo i membri del sistema per una matrice di permutazione elementare:

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad PAx = Pb \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

**Teorema.** Sia  $A$  una matrice  $m \times n$ . Esiste una matrice di permutazione  $P$   $m \times m$  tale che

$$PA = LR$$

ove  $L$  è una matrice  $m \times m$  triangolare inferiore con 1 sulla diagonale e  $R$  è una matrice  $m \times n$  trapezoidale superiore, tale che  $\text{rango}(A) = \text{rango}(R)$ .

Se  $A$  è quadrata non singolare, anche  $R$  è quadrata delle stesse dimensioni, non singolare.

La dimostrazione è costruttiva.

Consideriamo la prima colonna di  $A$ . Se c'è un elemento  $a_{r_1} \neq 0$ , si premoltiplica  $A$  per una matrice di permutazione elementare  $P_1$  che

scambia le righe  $r$  e 1 per portare l'elemento in posizione perno e poi si esegue una trasformazione elementare di Gauss  $L_1$  che annulla tutti gli elementi della prima colonna al di sotto dell'elemento diagonale. Se al contrario tutta la prima colonna è nulla e, in tal caso, si pone  $P_1 = L_1 = I$ .

$$L_1 P_1 A_1 = A_2 \quad A \equiv A_1$$

Si cerca un elemento non nullo sulla seconda colonna dalla posizione di riga 2 alla riga  $n$ . Se esiste tale elemento, si porta in posizione perno, scambiando la seconda riga con la riga in cui sta l'elemento (mediante la permutazione  $P_2$ ) e poi si esegue una trasformazione di Gauss  $L_2$  per annullare tutti gli elementi al di sotto della posizione perno. Altrimenti si pone  $L_2 = P_2 = I$  e si prosegue.

Dopo  $k = \min(m - 1, n)$  passi si ottiene

$$L_k P_k \dots L_1 P_1 A = R$$

ove  $R$  è una matrice  $m \times n$ ,  $R = A_{k+1}$ .

Se  $m \leq n$ ,  $k = m - 1$ ,

$$R = \begin{pmatrix} \backslash & \dots & \dots & \dots \\ & \backslash & \dots & \dots \\ & & \backslash & \dots \\ & & & \dots \end{pmatrix}$$

Se  $m > n$ ,  $k = n$ ,

$$R = \begin{pmatrix} \backslash & \dots & \dots \\ & \backslash & \dots \\ & & \backslash \\ & & & \dots \\ & & & & \dots \\ & & & & & \dots \\ & & & & & & \dots \end{pmatrix}$$

Gli elementi diagonali di  $R$  sono nulli in corrispondenza dei perni nulli.

Per esempio, se  $A$  è quadrata di ordine  $n$  non singolare, esiste un elemento diverso da 0 sulla prima colonna (altrimenti ci sarebbe una colonna nulla) e pertanto si esegue una permutazione per portarlo in posizione perno e una trasformazione di Gauss:

$$L_1 P_1 A = \begin{pmatrix} a_{11}^{(2)} & \dots & \dots & a_{1n}^{(2)} \\ 0 & \dots & \dots & \dots \\ 0 & & \tilde{A}_2 & \dots \\ 0 & \dots & \dots & \dots \end{pmatrix} = A_2$$

Ora nella prima colonna di  $\tilde{A}_2$  esiste almeno un elemento non nullo, altrimenti  $\det(A_2) = 0$ . Ma  $A_2$  è prodotto di matrici non singolari e, quindi, è non singolare.

Al passo  $j$ ,

$$L_{j-1} P_{j-1} \dots L_1 P_1 A = A_j = \begin{pmatrix} a_{11}^{(j)} & \dots & \dots & a_{1n}^{(j)} \\ 0 & \backslash & \dots & \dots \\ 0 & & a_{jj}^{(j)} & \dots \\ 0 & & \dots & \tilde{A}_j \end{pmatrix}$$

$A_j$  è non singolare perchè prodotto di matrici non singolari. Uno degli elementi della prima colonna di  $\tilde{A}_j$  è non nullo, poichè altrimenti  $\det(A_j) = 0$ .

Pertanto in  $k = n - 1$  passi si ottiene

$$L_{n-1} P_{n-1} \dots L_1 P_1 A = R$$

con  $R$  non singolare.

In generale

$$L_k P_k \dots L_2 P_2 L_1 P_1 A = R$$

Si pone

$$S_{k-j} = P_k P_{k-1} \dots P_{k-j}$$

$$S_{k-j-1} = S_{k-j} P_{k-j-1}$$

$S_{k-j}$  è invertibile e  $S_{k-j}^{-1} = P_{k-j} \dots P_k$ . Allora,

$$L_k P_k L_{k-1} P_{k-1} L_{k-2} P_{k-2} \dots L_3 P_3 L_2 P_2 L_1 P_1 A = R$$

$$L_k P_k L_{k-1} S_k^{-1} S_k P_{k-1} L_{k-2} S_{k-1}^{-1} S_{k-1} P_{k-2} \dots$$

$$\dots L_3 S_4^{-1} S_4 P_3 L_2 S_3^{-1} S_3 P_2 L_1 S_2^{-1} S_2 P_1 A = R$$

$$L_k (S_k L_{k-1} S_k^{-1}) (S_{k-1} L_{k-2} S_{k-1}^{-1}) (S_{k-2} L_{k-2} S_{k-2}^{-1}) \dots$$

$$\dots (S_4 L_3 S_4^{-1}) (S_3 L_2 S_3^{-1}) (S_2 L_1 S_2^{-1}) S_1 A = R$$

Si pone  $P = S_1$  e si osserva che

$$\begin{aligned} S_i L_{i-1} S_i^{-1} &= S_i (I - m^{(i-1)} e_{i-1}^T) S_i^{-1} = I - S_i m^{(i-1)} e_{i-1}^T S_i^{-1} \\ &= I - \tilde{m}^{i-1} e_{i-1}^T = \tilde{L}_{i-1} \end{aligned}$$

$\tilde{L}_{i-1}$  ha la stessa struttura di  $L_{i-1}$ . Infatti  $S_i = P_k \dots P_i$  permuta elementi che stanno dalla posizione  $i$  a posizioni superiori.

$$S_i m^{(i-1)} = \tilde{m}^{(i-1)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ x \\ x \end{pmatrix} \quad i - 1 \text{ zeri} \quad e_{i-1}^T S_i^{-1} = e_{i-1}^T$$

Allora

$$L_k \tilde{L}_{k-1} \tilde{L}_{k-2} \dots \tilde{L}_1 P A = R$$

$$PA = \underbrace{\tilde{L}_1^{-1} \dots \tilde{L}_{k-1}^{-1} L_k^{-1}}_L R$$

$$L = \begin{pmatrix} 1 & & & \\ \dots & 1 & & \\ \dots & m_{ij} & 1 & \\ \dots & \dots & \dots & 1 \end{pmatrix}$$

Gli  $m_{ij}$  sono permutati di righe.

Ora, si dimostra che, se  $\text{rango}(A) = r$ ,  $\text{rango}(R) = r$ , ossia  $R$  ha solo  $r$  elementi diagonali non nulli.

Infatti  $\text{rango}(A) = r \Leftrightarrow$  ha  $r$  colonne al più linearmente indipendenti (siano  $i_1, \dots, i_r$ )  $\Leftrightarrow \lambda_1 A_{*i_1} + \lambda_2 A_{*i_2} + \dots + \lambda_r A_{*i_r} = 0 \quad \lambda_i = 0$

$$\Leftrightarrow \lambda_1 (P^T L R_{*i_1}) + \dots + \lambda_r (P^T L R_{*i_r}) = 0 \quad \lambda_i = 0$$

$$\Leftrightarrow P^T L (\lambda_1 R_{*i_1} + \dots + \lambda_r R_{*i_r}) = 0 \quad \lambda_i = 0$$

$$\Leftrightarrow \lambda_1 R_{*i_1} + \dots + \lambda_r R_{*i_r} = 0 \quad \lambda_i = 0$$

$$\Leftrightarrow \text{rango}(R) = r$$

Il sistema da risolvere  $Ax = b$  diventa, con  $A$  non singolare, poichè  $PA = LR$ :

$$\begin{cases} Ly = Pb \\ Rx = y \end{cases}$$

Inoltre vale che:

$$\det(A) = (-1)^\sigma r_{11} \dots r_{nn}$$

ove  $\sigma$  è il numero di permutazioni effettivamente effettuate sulla ennupla  $(1, 2, \dots, n)$ . Infatti il determinante di ogni matrice di permutazione elementare diversa dall'identità vale -1.

ESEMPIO.

$$\left( \begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{array} \right)$$

$$P_1 = I \quad L_1 = \left( \begin{array}{ccc} 1 & & \\ -1 & 1 & \\ -1 & 0 & 1 \end{array} \right) \quad L_1 P_1 [A \ b] = \left( \begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array} \right)$$

$$P_2 = \left( \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{array} \right) \quad L_2 = I$$

$$L_2 P_2 L_1 P_1 [A \ b] = \left( \begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right)$$

$$R = \left( \begin{array}{ccc} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array} \right) \quad L^{-1} P b = \left( \begin{array}{c} 1 \\ 0 \\ 1 \end{array} \right)$$

$$\Rightarrow x_3 = 1 \quad x_2 = -1 \quad x_1 = 1$$

$$P = P_2 P_1 = \left( \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{array} \right) \quad L^{-1} = L_2 P_2 L_1 P_2^T = L_2 \tilde{L}_1$$

$$L = \tilde{L}_1^{-1} I = \left( \begin{array}{ccc} 1 & & \\ 1 & 1 & \\ 1 & 0 & 1 \end{array} \right)$$

$$PA = LR$$

$$\left( \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 1 & 2 \end{array} \right) = \left( \begin{array}{ccc} 1 & & \\ 1 & 1 & \\ 1 & 0 & 1 \end{array} \right) \left( \begin{array}{ccc} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array} \right)$$

La presenza di un perno nullo è causa d'arresto nell'algoritmo di eliminazione di Gauss. Occorre ricorrere alla tecnica di ricerca del perno. Cosa accade in presenza di un perno piccolo?

Si consideri:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$[A_2 \ b_2] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0.0001 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Teoricamente si può procedere poichè  $a_{22}^{(2)} \neq 0$ :

$$m_{32} = \frac{1}{0.0001} = 10^4 \quad [A_3 \ b_3] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0.0001 & 1 & 1 \\ 0 & 0 & -9999 & -10^4 \end{pmatrix}$$

Le soluzioni esatte sono  $x_3 = 1.0001\dots$ ,  $x_2 = -1.0001\dots$ ,  $x_1 = 1$ .

Usando aritmetica finita con  $t = 4$  cifre decimali:

$$\tilde{x}_3 = 1.000$$

$$\tilde{x}_2 = (1 - 1)/10^{-4} = 0$$

$$\tilde{x}_1 = (1 - 0 - 1)/1 = 0$$

Gli errori in  $\tilde{x}_2$  e  $\tilde{x}_1$  sono grandi. Si noti che il perno  $a_{22}^{(2)}$  è piccolo ( $10^{-4}$ ). Se c'è un piccolo errore nella determinazione di  $\tilde{x}_3$ , questo viene amplificato di  $10^4$  nel calcolo di  $\tilde{x}_2$  e, di conseguenza, si ripercuote nella determinazione di  $\tilde{x}_1$ . Allora occorre evitare che i perni siano molto piccoli (e i moltiplicatori grandi).

$$\begin{aligned}
 & \left( \begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 1 & 1.0001 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{array} \right) \\
 [A_2 \ b_2] &= \left( \begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 0.0001 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array} \right)
 \end{aligned}$$

Si può sfruttare la libertà di scelta del perno per rendere l'algoritmo più stabile (ossia meno sensibile agli errori di arrotondamento delle operazioni).

$$P_2[A_2 \ b_2] = \left( \begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0.0001 & 1 & 1 \end{array} \right)$$

$$m_{32} = 10^{-4} \quad a_{33}^{(3)} = 1 - 10^{-4} = .9999 \quad b_3^{(3)} = 1$$

$$(R \ y) = \left( \begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & .9999 & 1 \end{array} \right)$$

Con  $t = 4$  cifre decimali,

$$\bar{x}_3 = 1.0000 \quad \bar{x}_2 = -1 \quad \bar{x}_1 = 1$$

Si noti che non c'è stato esagerato accrescimento nei valori di  $R$ , poichè il moltiplicatore è minore di 1.

Calcolo dei residui.

$$\tilde{r} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \bar{r} = \begin{pmatrix} 0 \\ 10^{-4} \\ 0 \end{pmatrix}$$

Con la strategia che sceglie il perno come l'elemento di modulo massimo sulla colonna, il residuo resta piccolo.

## Strategia di pivoting parziale

Si sceglie come perno l'elemento  $a_{rk}^{(k)}$  tale che

$$|a_{rk}^{(k)}| = \max_{i=k,n} |a_{ik}^{(k)}| \quad k = 1, 2, \dots, n - 1$$

In tal modo i moltiplicatori  $|m_{ik}| \leq 1$ ,  $i = k + 1, \dots, n$ ;  $k = 1, 2, \dots, n - 1$ .

Con questa strategia, è garantito che il residuo è piccolo.

Attenzione! Ciò non implica che la soluzione sia accettabile:

$$\left( \begin{array}{cc|c} 0.780 & 0.563 & 0.217 \\ 0.913 & 0.659 & 0.254 \end{array} \right)$$

Se si applica il pivoting parziale con  $t = 3$  cifre decimali:

$$m_{21} = \frac{0.780}{0.913} = 0.854 \quad \left( \begin{array}{cc|c} 0.913 & 0.659 & 0.254 \\ 0 & 0.001 & 0.001 \end{array} \right)$$

$$\tilde{x}_2 = 1; \tilde{x}_1 = -0.443$$
$$\tilde{r} = \begin{pmatrix} -0.000460 \\ -0.000541 \end{pmatrix} \quad \|\tilde{r}\|_\infty < 10^{-3}$$

Ma la soluzione esatta è  $x = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ .

Questo è un problema di mal condizionamento che non si risolve anche se si usa un algoritmo stabile.

Conclusione: la libertà di scelta del perno è sfruttata per dare maggiore stabilità all'algoritmo, limitando l'amplificarsi degli errori di arrotondamento nelle operazioni.

Esempio. Si usa  $t = 3, \beta = 10$ .

$$A = \begin{pmatrix} .58 & -1.1 & -0.52 \\ -.56 & 1.12 & .56 \\ .02 & .02 & .04 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} .58 & -1.1 & -0.52 \\ & .06 & .058 \\ & & .0019 \end{pmatrix}$$

Se si pone  $\epsilon = 10^{-3}$ ,  $r_{ii} > \epsilon$  e  $\text{rango}(A) = 3$ .

Se si pone  $\epsilon = 2 \cdot 10^{-3}$ ,  $r_{33} < \epsilon$  e  $\text{rango}(A) = 2$ .

Se si pone  $\epsilon = 10^{-1}$ ,  $\text{rango}(A) = 1$ . Una piccola variazione di  $\epsilon$  può generare una grande variazione del numero di elementi che si assumono nulli. Si parla di **pseudorange numerico** di una matrice.

## Strategia di pivoting totale

Al passo  $k$  si sceglie come perno l'elemento di modulo massimo della sottomatrice  $\tilde{A}_k$ .

$$|a_{ij}^{(k)}| = \max_{r,s=k,n} |a_{rs}^{(k)}| \quad k = 1, 2, \dots, n - 1$$

Ciò richiede di eseguire lo scambio tra la riga  $i$  e la riga  $k$  e la colonna  $j$  e la colonna  $k$ .

Questo comporta un riordinamento delle soluzioni dopo aver eseguito l'algoritmo.

Sono richiesti  $\mathcal{O}(\frac{n^3}{3})$  confronti, anche se l'algoritmo è più stabile.

L'algoritmo si basa sul seguente teorema generale.

**Teorema.** Sia  $A$  una matrice  $m \times n$  di rango  $r$ . Allora esistono due matrici di permutazione  $P$  e  $Q$  di ordine  $m$  e  $n$  rispettivamente tali che:

$$PAQ = LR$$

ove  $L \in \mathbb{R}^{m \times m}$  triangolare inferiore unitaria e  $R$  trapezoidale superiore  $m \times n$  di rango  $r$ , avente esattamente  $r$  righe che al di sopra della diagonale principale compresa possono contenere elementi non nulli:

$$PAQ = L \begin{pmatrix} \backslash & \dots & \dots \\ & \backslash & \dots \\ & & \dots \end{pmatrix} \quad r \text{ righe}$$

$$\text{rango}(A) = r$$

Se  $A$  è non singolare,  $R$  è triangolare superiore non singolare.

Esempio.

$$A = A_1 = \begin{pmatrix} 1 & 1 & 1 & 4 & 1 \\ -2 & -1 & 0 & 1 & 3 \\ -1 & 0 & 1 & 1.7 & 4 \\ 1 & 1.4 & 1.8 & 1 & 3 \\ 0 & 1 & 2 & 3 & 5 \end{pmatrix}$$

$$A_2 = L_1 P_1 A Q_1 = \begin{pmatrix} 5 & 1 & 2 & 3 & 0 \\ 0 & -1.6 & -1.2 & -.8 & -2 \\ 0 & -.8 & -.6 & -.7 & -1 \\ 0 & .8 & .6 & -.8 & 1 \\ 0 & .8 & .6 & 3.4 & 1 \end{pmatrix}$$

$$A_3 = L_2 P_2 L_1 P_1 A Q_1 Q_2 = \begin{pmatrix} 5 & 3 & 2 & 1 & 0 \\ 0 & 3.4 & 0.6 & 0.8 & 1 \\ 0 & 0 & -0.4765 & -0.6353 & -0.7941 \\ 0 & 0 & 0.7412 & 0.9882 & 1.2353 \\ 0 & 0 & -1.0588 & -1.4118 & -1.7647 \end{pmatrix}$$

$$A_4 = L_3 P_3 L_2 P_2 L_1 P_1 A Q_1 Q_2 Q_3 =$$

$$= \begin{pmatrix} 5 & 3 & 2 & 1 & 0 \\ 0 & 3.4 & 0.6 & 0.8 & 1 \\ 0 & 0 & -1.7647 & -1.4118 & -1.0588 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$\text{rango}(R) = 3$ . In aritmetica finita, si deduce uno pseudorango numerico che non è detto che coincida con il rango teorico e dipende dalla tolleranza scelta.

## Riduzione in scala

Invece del pivoting totale (troppo costoso), si può usare la tecnica di riduzione in scala del sistema

$$Ax = b$$

Si tratta di costruire un sistema equivalente al dato mediante l'uso di matrici diagonali (e quindi facilmente invertibili) in modo da abbassare il numero di condizionamento del sistema:

$$D_1 A D_2 D_2^{-1} x = D_1 b$$

$$A' y = d$$

con  $A' = D_1 A D_2$ ,  $D_2^{-1} x = y$  e  $d = D_1 b$ . Si vuole che  $K(A) < K(A')$ , ove con  $K(A)$  si intende il numero di condizione della matrice  $A$ . Si tratta di determinare  $D_1$  e  $D_2$  diagonali in modo da abbattere il condizionamento del sistema. E' un problema complesso. Si possono usare varie tecniche.

**Prima Idea.** Fare in modo che la matrice trasformata sia tale che

$$\max_{j=1,n} |a'_{ij}| = 1$$

$i = 1, \dots, n$ ; in tal caso si pone  $D_1 = \text{diag}(d_1^{(1)}, \dots, d_n^{(1)})$ ,  $D_2 = I$ , con  $d_i^{(1)} = \frac{1}{\max_{j=1,n} |a_{ij}|}$  (scalatura per righe)

oppure

$$\max_{i=1,n} |a'_{ij}| = 1$$

$j = 1, \dots, n$ ; in tal caso si pone  $D_2 = \text{diag}(d_1^{(2)}, \dots, d_n^{(2)})$ ,  $D_1 = I$ , con  $d_j^{(2)} = \frac{1}{\max_{i=1,n} |a_{ij}|}$  (scalatura per colonne).

L'idea è quella di calcolare  $d_i^{(1)}$  (o  $d_i^{(2)}$ ) a priori e, senza effettivamente eseguire la scalatura sulla matrice, calcolare il pivot come l'elemento  $a_{rk}^{(k)}$  tale che

$$\frac{|a_{rk}^{(k)}|}{\max_{j=1,n} |a_{rj}|} = |a_{rk}^{(k)}| d_r^{(1)} = \max_{i=k,n} |a_{ik}^{(k)}| \cdot d_i^{(1)}$$

**Seconda Idea.** Si effettua la scalatura per righe (per colonne) in modo che

$$\sum_{j=1}^n |a'_{ij}| = 1 \quad \left( \sum_{i=1}^n |a'_{ij}| = 1 \right)$$

ponendo  $d_i^{(1)} = \frac{1}{\sum_{j=1}^n |a_{ij}|}$  ( $d_i^{(2)} = \frac{1}{\sum_{i=1}^n |a_{ij}|}$ ).

La scalatura altera la scelta del pivot. Non è detto, tuttavia, che ciò comporti un miglioramento significativo nella soluzione.

Esempio.

$$\left( \begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 1 & -1 & 1 & 1 \\ 2 \cdot 10^{-4} & 1 & 1 & 2 \end{array} \right)$$

La soluzione esatta è  $x = (1.0010\dots, 1.0004\dots, .9994\dots)^T$ .

Supponendo di lavorare con aritmetica decimale con  $t = 3$  cifre di precisione, si ha:

$$\left( \begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & -3 & -2 & -5 \\ 0 & .999 & .999 & 1.99 \end{array} \right) \rightarrow \left( \begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 0 & -3 & -2 & -5 \\ 0 & 0 & .333 & .325 \end{array} \right)$$

La soluzione trovata è  $\tilde{x} = (.975, 1.01, 1.05)^T$ .

Scalando per righe (prima idea), si calcolano  $d_1^{(1)} = 1/3$ ,  $d_2^{(1)} =$

1,  $d_3^{(1)} = 1$ . Allora, si ottiene:

$$\left( \begin{array}{ccc|c} 1/3 & 2/3 & 1 & 2 \\ 1 & -1 & 1 & 1 \\ 2 \cdot 10^{-4} & 1 & 1 & 2 \end{array} \right)$$

Di conseguenza, viene alterata la scelta del pivot:

$$\left( \begin{array}{ccc|c} 1 & -1 & 1 & 1 \\ .333 & .666 & 1 & 2 \\ 2 \cdot 10^{-4} & 1 & 1 & 2 \end{array} \right) \rightarrow \left( \begin{array}{ccc|c} 1 & -1 & 1 & 1 \\ 0 & .999 & .666 & 1.66 \\ 0 & 1 & .999 & 1.99 \end{array} \right)$$

$$\rightarrow \left( \begin{array}{ccc|c} 1 & -1 & 1 & 1 \\ 0 & .999 & .666 & 1.66 \\ 0 & 0 & -.332 & -.328 \end{array} \right)$$

La soluzione trovata è  $\bar{x} = (.987, 1.00, 1.01)^T$ .

Schwarz suggerisce di non eseguire effettivamente la scalatura, ma di scegliere come pivot  $a_{rk}^{(k)}$  tale che

$$\frac{|a_{rk}^{(k)}|}{\sum_{j=k}^n |a_{rj}^{(k)}|} = \max_{i=k,n} \frac{|a_{ik}^{(k)}|}{\sum_{j=k}^n |a_{ij}^{(k)}|}$$

e di ripetere tale scalatura ad ogni passo. Questa tecnica, più costosa del semplice metodo di Gauss con pivoting parziale fornisce buoni risultati.

Esempio

$$\left( \begin{array}{ccc|c} 1 & 2 & 3 & 6 \\ 1 & -1 & 1 & 1 \\ 2 \cdot 10^{-4} & 1 & 1 & 2 \end{array} \right)$$

La soluzione esatta è  $x = (1.0010\dots, 1.0004\dots, .9994\dots)^T$ . Si pone  $d_1 = 1/6, d_2 = 1/3, d_3 = .499$ . La scelta del pivot è  $a_{21}$ :

$$\left( \begin{array}{ccc|c} 1 & -1 & 1 & 1 \\ 1 & 2 & 3 & 6 \\ 2 \cdot 10^{-4} & 1 & 1 & 2 \end{array} \right) \rightarrow \left( \begin{array}{ccc|c} 1 & -1 & 1 & 1 \\ 0 & 3 & 2 & 5 \\ 0 & 1 & .999 & 1.99 \end{array} \right)$$

Ora abbiamo  $d_2 = 1/5, d_3 = .500$ . Il pivot resta  $a_{22}^{(2)}$ :

$$\left( \begin{array}{ccc|c} 1 & -1 & 1 & 1 \\ 0 & 3 & 2 & 5 \\ 0 & 0 & .333 & .330 \end{array} \right)$$

La soluzione calcolata è  $\hat{x} = (.990, 1.00, 1.01)^T$

## Matrici a banda

Si dice che  $A$  è una matrice a banda con banda superiore  $s$  e banda inferiore  $r$  se  $a_{ij} = 0, j - i > s, i - j > r$ .

$$A = \begin{pmatrix} a_{11} & \dots & a_{1s+1} & & & \\ \dots & \dots & \dots & a_{2s+2} & & \\ a_{r+11} & \dots & \dots & \dots & \dots & \\ & a_{r+22} & \dots & \dots & \dots & \\ & & \dots & \dots & \dots & a_{nn} \end{pmatrix}$$

Se non è necessario pivoting, la fattorizzazione  $A = LR$  produce una matrice  $L$  di banda inferiore  $r$  e una matrice  $R$  di banda superiore  $s$ . Si ha dunque minore occupazione di memoria e minore complessità computazionale.

$$L = \begin{pmatrix} 1 & & & & & \\ \dots & 1 & & & & \\ l_{r+11} & \dots & 1 & & & \\ & l_{r+22} & & 1 & & \\ & & & \dots & \dots & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} r_{11} & \dots & r_{1s+1} & & & \\ & \dots & \dots & r_{2s+2} & & \\ & & \dots & \dots & \dots & \\ & & & \dots & \dots & \\ & & & & & r_{nn} \end{pmatrix}$$

Se è necessario pivoting parziale,  $L$  mantiene banda  $r$ , ma  $R$  ha banda superiore  $2s$ . Il pivoting parziale distrugge la struttura della matrice.

## Matrici tridiagonali

Si consideri una matrice  $A$  tridiagonale strettamente diagonale dominante oppure non singolare diagonale dominante o definita positiva. Tre vettori  $c, d, b$  sono sufficienti a memorizzare gli elementi non nulli delle tre diagonali di  $A$ . Supponiamo che si debba risolvere il sistema

$$Ax = f$$

$$A = \begin{pmatrix} d_1 & b_1 & & & & \\ c_2 & d_2 & b_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & b_{n-1} \\ & & & & c_n & d_n \end{pmatrix}$$

ove  $c_2 = b_n = 0$ . Poichè  $A$  è a banda con banda superiore e inferiore 1,  $L$  è triangolare inferiore unitaria con banda inferiore 1 e  $R$  è triangolare superiore a banda superiore 1.

$$A = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & \ddots & 1 & & \\ & & \ddots & 1 & \\ & & & l_n & 1 \end{pmatrix} \begin{pmatrix} u_1 & s_1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & s_{n-1} & \\ & & & & u_n \end{pmatrix} = LR$$

Si osservi che:

$$b_i = a_{ii+1} = (0 \dots l_i \ 1 \ 0 \ \dots) \begin{pmatrix} \vdots \\ 0 \\ s_i \\ u_{i+1} \\ 0 \\ \vdots \end{pmatrix} = s_i \quad i = 1, \dots, n - 1$$

$$c_i = a_{ii-1} = (0 \dots l_i \ 1 \ 0 \ \dots) \begin{pmatrix} \vdots \\ 0 \\ b_{i-2} \\ u_{i-1} \\ 0 \\ \vdots \end{pmatrix} = l_i u_{i-1} \quad i = 2, \dots, n$$

$$\Rightarrow l_i = \frac{c_i}{u_{i-1}} \quad i = 2, \dots, n$$

$$\left\{ \begin{array}{l} d_1 = u_1 \\ d_i = a_{ii} = (0 \dots l_i \ 1 \ 0 \ \dots) \begin{pmatrix} \vdots \\ 0 \\ b_{i-1} \\ u_i \\ 0 \\ \vdots \end{pmatrix} = l_i b_{i-1} + u_i \quad i = 2, \dots, n \end{array} \right.$$

$$\Rightarrow u_1 = d_1; \quad u_i = d_i - l_i b_{i-1} \quad i = 2, \dots, n$$

I due sistemi triangolare inferiore e superiore si risolvono nel seguente modo:

$$f_i = f_i - l_i f_{i-1}, \quad i = 2, \dots, n$$

$$f_n = f_n / u_n; \quad f_i = (f_i - b_i f_{i+1}) / u_i \quad i = n - 1, \dots, 1$$

### Complessità computazionale

- Fattorizzazione:  $n - 1$  divisioni,  $n - 1$  somme e  $n - 1$  prodotti;
- Soluzione:  $n$  divisioni,  $2(n - 1)$  somme,  $2(n - 1)$  prodotti.

Anche nel caso di matrici di Hessemberg, l'applicazione del metodo di Gauss (anche con pivoting) comporta un abbassamento della complessità computazionale. Infatti in totale si eseguono  $n - 1$  divisioni per calcolare i moltiplicatori e  $\mathcal{O}(n^2/2)$  prodotti e altrettante somme per il calcolo degli elementi di  $R$ .

Per matrici sparse, se non si esegue un riordinamento delle righe e colonne, si creano dei riempimenti, detti **fill-in**, che distruggono la struttura della matrice.

Esempio. Il fattore di Cholesky della matrice sparsa  $A$  è denso.

$$A = \begin{pmatrix} 4 & 1 & 2 & 1/2 & 2 \\ 1 & 1/2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 1/2 & 0 & 0 & 5/8 & 0 \\ 2 & 0 & 0 & 0 & 16 \end{pmatrix}$$

↓

$$\mathcal{L} = \begin{pmatrix} 2 & & & & & \\ 0.5 & 0.5 & & & & \\ 1 & -1 & 1 & & & \\ 0.25 & -0.25 & -0.5 & 0.5 & & \\ 1 & -1 & -2 & -3 & 1 & \end{pmatrix}$$

Esistono tecniche che riordinano una matrice per minimizzare il fill-in. Una delle più note è il criterio di Markowitz: seguendo questo criterio, ad ogni passo  $k$  si sceglie come perno l'elemento della sottomatrice  $\tilde{A}_k$  per cui la seguente quantità è minima:

$$(row_i - 1)(col_j - 1)$$

ove  $row_i$  e  $col_j$  sono i numeri di elementi non nulli sulla riga  $i$ -esima e sulla  $j$ -esima colonna di  $\tilde{A}_k$ .

Se una matrice è simmetrica, il criterio di Markowitz equivale al **minimum degree ordering**. Quest'ultima tecnica consiste nel costruire il grafo associato alla matrice: se  $n$  è la dimensione della matrice, si considera un grafo di  $n$  nodi numerati da 1 a  $n$  e poi, per ogni elemento  $a_{ij} \neq 0$  si genera un arco che connette il nodo  $i$  al nodo  $j$ .

Si prende come ordinamento quello dal nodo di grado minimo (grado di un nodo=numero di archi che partono da tale nodo) a quelli di grado superiore. Questo è dovuto al fatto che quando si elimina un nodo, tutti i nodi a lui connessi si connettono tra di loro, riflettendo la creazione di elementi non nulli. Pertanto si scelgono i nodi con minori connessioni prima degli altri.



