Résolution de systèmes

I. Matrice triangulaire

But : $Ax = b$

1. Triangulaire inférieure

$$L = \begin{pmatrix} L_{11} & & 0 \\ \dots & \ddots & \\ L_{n1} & \dots & L_{nn} \end{pmatrix}$$

a. Par ligne

$$x_i = \frac{b_i - \left(L_{i1}x_1 + \cdots L_{ij}x_j + \cdots + L_{ii-1}x_{i-1}\right)}{L_{ii}}$$

```
fonction b = tri_inf(L, b)
pour i = 1 à n faire
    pour j = 1 à i - 1 faire
        b(i) = b(i) - L(i,j)*b(j)
    b(i) = b(i)/L(i,i)

pour i = 1 à n faire
    b(i) = (b(i) - L(i,1:i-1)b(1:i-1))/L(i,i)
```

b. Par colonne

2. Triangulaire supérieure

$$U = \begin{pmatrix} U_{11} & \dots & U_{1n} \\ & \ddots & \vdots \\ 0 & & U_{nn} \end{pmatrix}$$

a. Par ligne

$$x_i = \frac{b_i - \left(U_{i,i+1}x_{i+1} + \cdots U_{ij}x_j + \cdots + U_{in}x_n\right)}{U_{ii}}$$

```
fonction b = tri_sup(U, b)
pour i = n à 1 pas de -1 faire
    pour j = i + 1 à n faire
        b(i) = b(i) - U(i,j)*b(j)
    b(i) = b(i)/U(i,i)

pour i = n à 1 pas de -1 faire
    b(i) = (b(i) - U(i,i+1:n)b(i+1:n))/U(i,i)
```

b. Par colonne

On calcule colonne par colonne les valeurs des x_j en mettant à chaque fois à jour les b_k suivants/précédents en y soustrayant $A_{kj}x_j$.

```
fonction b = tri_inf_c(L, b)
pour j = 1 à n faire
    b(j) = b(j)/L(j,j)
    b(j+1:n) = b(j+1:n) - b(j)L(j+1:n,j)
```

c. Version matricielle par ligne

```
pour i = 1 à n faire
    b = Ni*b
```

$$N_i = I - e_i v_i^T \quad v_i^T = \left(\frac{L(i, 1:i-1)}{L_{ii}}, 1 - \frac{1}{L_{ii}}, \underbrace{0, \dots, 0}_{n-i\text{ fois}}\right)$$

d. Version matricielle par colonne

```
pour j = 1 à n faire
    b = Mj*b
```

$$M_j = I - \tau_j e_j^T \quad \tau_j = \left(\underbrace{0; \dots; 0}_{j-1\text{ fois}}; 1 - \frac{1}{L_{jj}}; \frac{L(j+1:n, j)}{L_{jj}}\right)$$

$$x = \underbrace{M_n \dots M_1}_{L^{-1}} b = L^{-1} b$$

```
fonction b = tri_sup_c(U, b)
pour j = n à 1 pas de -1 faire
    b(j) = b(j)/U(j,j)
    b(1:j-1) = b(1:j-1) - b(j)U(1:j-1,j)
```

c. Version matricielle par ligne

```
pour i = n à 1 pas de -1 faire
    b = Ni*b
```

$$N_i = I - e_i v_i^T \quad v_i^T = \left(\underbrace{0, \dots, 0}_{i-1\text{ fois}}, 1 - \frac{1}{U_{ii}}, \frac{U(i, i+1:n)}{U_{ii}}\right)$$

d. Version matricielle par colonne

```
pour j = n à 1 pas de -1 faire
    b = Mj*b
```

$$M_j = I - \tau_j e_j^T \quad \tau_j = \left(\frac{U(1:j-1, j)}{U_{jj}}; 1 - \frac{1}{U_{ii}}; \underbrace{0; \dots; 0}_{n-j\text{ fois}}\right)$$

$$x = \underbrace{M_n \dots M_1}_{U^{-1}} b = U^{-1} b$$

II. Matrice diagonale

$$x_i = b_i / A_{ii}$$

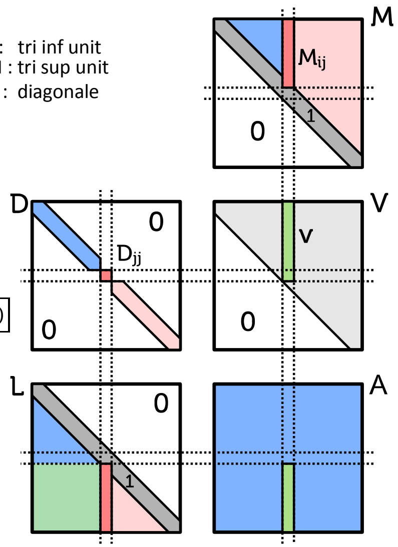
```
pour i = 1 à n faire
    si A(i,i) = 0
        si b(i) = 0
            Infinité de solution
        sinon
            Pas de solution
    sinon
        x(i) = b(i) / A(i,i)
```

Factorisation LDM et de Cholesky

III. Factorisation LDM

$$A = LV = L(DD^{-1})V = LD\underbrace{(D^{-1}V)}_{M} = L\underbrace{DM}_{V} \qquad D_{ii} = V_{ii}$$

L : tri inf unit
M : tri sup unit
D : diagonale



1. Maths

$\boxed{L(1{:}j,1{:}j)v = A(1{:}j,j)}$ $\quad v = V(1{:}j,j)$

$\boxed{D(j,j) = v(j)}$ $\quad \boxed{M(i,j) = \dfrac{v(i)}{D(i,i)}}$

$$\boxed{L(j+1{:}n,j) = \big(A(j+1{:}n,j) - L(j+1{:}n,{:}j-1)v(1{:}j-1)\big)/v(j)}$$

2. Codage

```
fonction [L,D,M] = factorisation_ldm(A)
pour j = 1 à n faire
    v(1:j) = tri_inf(L(1:j, 1:j), A(1:j,j))
    D(j,j) = v(j)
    pour i = 1 à j - 1 faire
        M(i,j) = v(i)/D(i,i)
    L(j+1:n,j) = A(j+1:n,j) - L(j+1:n,1:j-1)v(1:j-1)/v(j)
```

3. Matrices symétriques : $A = LDL^T$

Si A est symétrique, $M = L^T$, donc on connait v sans résoudre de système.

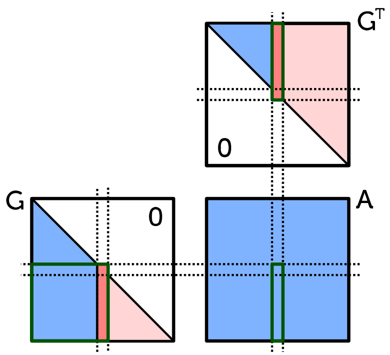
$\boxed{\underset{1\le i\le j-1}{v(i)} = D(i,i)L(j,i)}$ $\quad \boxed{v(j) = A(j,j) - L(j,1{:}j-1)v(1{:}j-1)}$

IV. Matrices définies positives

$$A \text{ définie positive} \Leftrightarrow \begin{cases} A \text{ symétrique} \\ \forall x \neq 0,\ \ x^T A x > 0 \end{cases}$$

V. Factorisation de Cholesky : $A = GG^T$

A définie positive et $A = LDL^T \quad \Rightarrow \quad D_{ii} > 0\ \forall i \Rightarrow \quad \underset{G\text{:tri.inf. unique}}{A = GG^T}$



1. Calcul

$$v = L(j,j)L(j{:}n,j) = A(j{:}n,j) - \sum_{k=1}^{j-1} L(j,k)L(j{:}n,k)$$

$$\Rightarrow \quad L(j,j)^2 = v(1) \quad \Rightarrow \quad \boxed{L(j{:}n,j) = v/\sqrt{v(1)}}$$

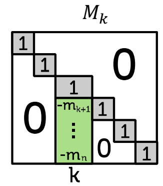
2. Codage

```
fonction A = Cholesky(A)
pour j = 1 à n faire
    si j > 1 alors
        pour k = 1 à j - 1 faire
            A(j:n,j) = A(j:n,j) - A(j,k)A(j:n,k)
    A(j:n,j) = A(j:n,j)/sqrt(A(j,j))
```

Factorisation LU

$Ax = b$ et $\underset{\substack{L\text{ : tri. inf. unit.}\\ U\text{ : tri. sup.}}}{A = LU} \quad \Rightarrow \quad L\underbrace{Ux}_{c} = b \quad \underline{Lc = b} \quad \underline{Lx = c}$ But : plus rapide

I. Factorisation LU

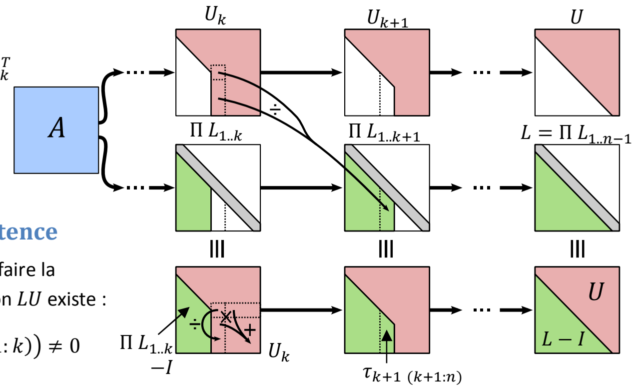


1. Transformation de Gauss

$$\underbrace{M_{n-1} \dots M_1}_{M} A = U \quad \Leftrightarrow \quad A = \underbrace{L_1 \dots L_{n-1}}_{M^{-1}=L} U = LU$$

$$\tau_k^T = \Big(\underbrace{0, \dots, 0}_{k \text{ fois}}, \frac{a_{k+1,k}}{a_{kk}}, \dots, \frac{a_{nk}}{a_{kk}}\Big)$$

$M_k = I - \tau_k e_k^T \quad L_k = I + \tau_k e_k^T$



2. Calcul de L

$$L_k L_{k'} = I + \tau_k e_k^T + \tau_{k'} e_{k'}^T$$
$$k < k'$$

3. Condition d'existence

Si à l'étape $a_{kk} = 0$, on ne pas faire la factorisation LU. La factorisation LU existe :

$$\Leftrightarrow \ \forall k \in [\![1; n-1]\!], \det\big(A(1{:}l, 1{:}k)\big) \neq 0$$

4. Algorithme

```
pour k = 1 à n-1 faire
    i = max |A(i,k)|     // PA=LU
    p(k) = i             // PA=LU
    A(k,:) ↔ A(i,:)      // PA=LU :permute U et les τ_k pour avoir les τ̃_k donc L̃

    A(k+1:n,k) = A(k+1:n,k)/A(k,k)
    └ τ_{k+1}(k+1:n)
    A(k+1:n,k+1:n) = A(k+1:n,k+1:n) - A(k+1:n,k) × A(k,k+1:n)
                                      └ τ_{k+1}(k+1:n)
```

```
pour k = 1 à n-1 faire
    pour i = k+1 à n faire
        A(i,k) = A(i,k)/A(k,k)        // τ_{k+1}(i)
    pour i = k+1 à n faire
        pour j = k+1 à n faire
            A(i,j) = A(i,j) - A(i,k) × A(k,j)        // U_k = M_k U_{k-1}
```

II. Factorisation $PA = LU$

$$M_{n-1}P^{(n-1)}M_{n-2}P^{(n-2)} \dots M_k P^{(k)} \dots M_1 P^{(1)} A = U$$

$$\Leftrightarrow \quad \underbrace{\widetilde{M}_{n-1} \dots \widetilde{M}_1}_{\widetilde{M}} \underbrace{P^{(n-1)} \dots P^{(1)}}_{P} A = U \quad \Leftrightarrow \quad PA = \underbrace{\tilde{L}_1 \dots \tilde{L}_{n-1}}_{\tilde{L} = \widetilde{M}^{-1}} U$$

$\boxed{\widetilde{M}_k = P^{(n-1)} \dots P^{(k+1)} M_k P^{(k+1)} \dots P^{(n-1)}}$ $\qquad \boxed{P_{ij} M_k P_{ij} = I - \tau_k^{(ij)} e_k^T}$