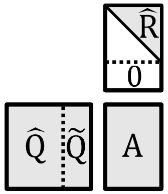
# Factorisation QR

## I. Factorisation QR

$A = QR = \hat{Q}\hat{R}$    $Q$ orthogonale, $R$ tri. sup.    $Q^TQ = QQ^T = \hat{Q}^T\hat{Q} = I \quad \hat{Q}\hat{Q}^T \neq I$




$\boxed{H_v = I - \frac{2}{\underbrace{v^Tv}_{\beta}} vv^T}$    Orthogonale, symétrique, $H_v = K_{\gamma v}$ $(\gamma \in \mathbb{R}^*)$

$H_v x = -\alpha e_1 \;\Rightarrow\; \boxed{v = x + \alpha e_1}$ et $\boxed{\alpha = \pm\|x\|}$ $\left(\begin{array}{ll}\text{US :} & -\|x\| \\ \text{Fr :} & signe(x_1)\,\|x\|\end{array}\right)$

$H_p \ldots H_1 A = R \;\Rightarrow\; \boxed{A = \underbrace{(H_1 \ldots H_p)^T}_{Q} R = QR}$

### Algorithmes

#### a. Calcul du vecteur de House

```
fonction [v, β] = vecteurDeHouse(x)

norm_x = x'*x
v = x                     % v = x + αe_1
v(1) = v(1) + ± √norm_x
β = 2 v(1)² / (norm_x + v(1)² - x(1)²)
v = v/v(1)
```

#### b. Produit House-vecteur

$$H_v x = \left(I - \frac{2}{v^Tv} vv^T\right) x = x - \frac{2}{\underbrace{v^Tv}_{\beta}} \underbrace{v^Tx}_{w}\; v$$

```
fonction x = House_vecteur(x, v, β)

w = v' * x
x = x - β * w * v
```

#### c. Transformation d'une matrice

```
fonction [A, β] = House_matrice(A)

[v, β] = vecteurDeHouse(A(:,1))  % calcul du vecteur de House
pour j = 1 à p faire             % calcul des produits House-vecteur
    A(:,j) = House_vecteur(A(:,j), v, β)
finpour
A(2:end,1) = v(2:end)            % stockage de v (sauf v(1) = 1)
```

#### d. Factorisation $QR$

```
fonction A = qr(A)

pour j = 1 à p faire
    [A(j:n, j:p), β] = House_matrice(A)
finpour
% A contient R et les vecteurs v
```

#### e. Calcul $Qb$

```
fonction b = Qb(A, b)

pour j = 1 à p faire
    % initialisation v
    v(j) = 1; v(j+1:n) = A(j+1:n,j)
    b(j:n) = b(j:n) - β_j * v * v' * b(j:n)
finpour
```

## II. Les moindres carrés pour résoudre $Bx = d$

$$\min_x \underbrace{\|Bx - d\|^2 = x^TAx - 2x^Tb}_{J(x)} \;\Leftrightarrow\; \boxed{\nabla_x J(x) = 2Ax - 2b = 0} \;\Leftrightarrow\; \boxed{Ax = b} \qquad \underline{A = B^TB} \quad \underline{b = B^Td}$$

*Formules de gradient :* $\nabla_x(x^Ta) = a \qquad \nabla_x(x^TAx) = Ax + A^Tx$

### Utilisation de $QR$

$$B = QR \quad A = R^TR \;\Leftrightarrow\; Ax - b = \hat{R}^T(\hat{R}x - \hat{Q}^Td) = 0 \;\Leftrightarrow\; \boxed{x = \hat{R}^{-1}\underbrace{\hat{Q}^Td}_{c}}$$

```
fonction x = moindres_carres_chol(B, d)

A = B'*B     B = A'*d     L = chol(A)
y = tri_inf(L, b)   x = tri_sup(L', y)
```

```
fonction x = moindres_carres_qr(B, d)

RetV = qr(B)         c = qr_b(RetV)
x = tri_sup(RetV, c)
```

# Calcul de valeurs propres

## I. Valeurs propres

$A \in \mathbb{R}^{p\times p} \rightarrow \quad \lambda_i \in \mathbb{C} \quad v_i \in \mathbb{C}^p$

$Av_i = \lambda_i v_i$

$AV = VD$

$v \rightarrow \lambda :\; \lambda = \frac{v^TAv}{v^Tv} = \frac{v^TAv}{\|v\|^2} \underset{\text{si } \|v\|^2=1}{=} v^TAv$

$\lambda \rightarrow v :\; (A - \lambda I)v = 0$

$R_i = \left\{ z \in \mathbb{C} \;|\; |z - a_{ii}| \leq \sum_{j\neq i} |a_{ij}| \right\}$

cercles de Gershgorin : contiennent tous les $\lambda$

## II. Valeurs singulières

$B \in \mathbb{R}^{n\times p} \;\rightarrow\; \mu_i \in \mathbb{R} \quad u_i \in \mathbb{R}^p \quad v_i \in \mathbb{R}^p$

$Bv_i = \mu_i u_i \qquad B^Tu_i = \mu_i v_i$

$B = UDV^T$

$A = B^TB \;\Rightarrow\; (\mu_i^2, v_i)$ val. et vect. p. de $A$

## III. Matrice carrée symétrique définie positive

$(\lambda_i, v_i) \in \mathbb{R} \times \mathbb{R}^p \qquad \lambda_1 \underset{\text{conv°}}{\geq} \lambda_2 \geq \cdots \geq 0 \qquad v_i^Tv_i \underset{\text{conv°}}{=} 1$

$v_i^Tv_j = 0 \quad V^TV = I$ (base de $\mathbb{R}^n$)

## IV. Méthode de la puissance itérée

$z^{(k+1)} = \frac{Az^{(k)}}{\|Az^{(k)}\|} \qquad \|Az^{(k)}\| \underset{k\to\infty}{\rightarrow} |\lambda_1| \qquad z^{(k)} \underset{k\to\infty}{\propto} v_1 \qquad A^{(2)} = A - \lambda_1 v_1 v_1^T$

Déflation : vp suivante

```
fonction [z, λ] = PuissanceIteree(A, z)

tant que (non conv) faire
    z = Az
    z = z/‖z‖
fintantque
λ = z'Az
```

```
fonction [Z, D] = PuissanceItereeMult(A, Z)

tant que (non conv) faire
    Z = AZ
    Z = orthogonalise(Z)      [Z R] = qr(Z)
fintantque
D = Z'AZ
```

## V. Calcul de valeurs propres avec $QR$

$$\begin{cases} Z_k = AQ_k \\ (Q_{k+1}, R_{k+1}) = qr(Z_k) \end{cases} \qquad T_k = Q_k^TAQ_k \rightarrow D$$

$$T_k = Q_k^TAQ_k = Q_k^TQ_{k+1}R_{k+1} = Q_{k+1}^{(2)}R_{k+1}$$
$$T_{k+1} = Q_{k+1}^TAQ_{k+1} = R_{k+1}Q_{k+1}^{(2)}$$

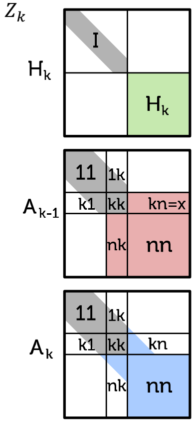
$\Rightarrow$ On a une suite de $T_k$ calculable via $QR$ pas besoin de $Z_k$

```
fonction [Z, D] = Ite_QR(A)

T = Q_0^T A Q_0
tant que (non conv) faire
    [Q R] = qr(T)
    T = RQ
fintantque
```

```
fonction [Z, D] = Ite_QR_tridiag(A)

T = tri_diag(A)
tant que (non conv) faire
    [Q R] = qr_tri(T)
    T = RQ
fintantque
```




### Tridiagonalisation

$A^{(k)} = H_k^TA^{(k-1)}H_k$

```
fonction A = tri_diag (A)

pour k = 1 à n-2 faire
    [v, β] = vecteurDeHouse(A(k+1:n,k))      % x = A_kn
    p = β * A(k+1:n,k+1:n) * v               % p = β A_nn v
    w = p - (β * v' * p / 2) * v             % w = p - (β v^T p / 2) v
    A(k+1,k) = A(k,k+1) = ‖A(k+1:n,k)‖       % A_nk^(k) = A_kn^(k)T = -αe_1
    A(k+1:n,k+1:n) = A(k+1:n,k+1:n) - v*w' - w*v'   % A_nn^(k) = H_k^T A_nn H_k = A_nn - vw^T - wv^T
finpour
```

# Méthodes itératives

## I. Les méthodes itératives

Cherche solution de $Ax = b$ avec $\left(x^{(k)}\right)_{k\in\mathbb{N}} \to \tilde{x}$

$$Ax = b \Leftrightarrow \sum_{j=1}^{i-1} A_{ij}x_j + A_{ii}\boldsymbol{x_i} + \sum_{j=i+1}^{n} A_{ij}x_j = b_i$$

**fonction** `x = iterative (A, ite_max, epsilon, x [, ω])`

**tant que** (non conv) **faire** $\quad n_{iter} < n_{iter_{max}}$ et $\left\|x^{(k+1)} - x^{(k)}\right\| > \varepsilon$ ou $\left\|Ax^{(k)} - b\right\| > \varepsilon$

*Forme matricielle* $(A = D + U + L)$

*Jacobi*

**pour** i = 1 **à** n **faire**

$\quad$ y(i) = $\dfrac{b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^{n} A_{ij}x_j}{A_{ii}}$

**finpour**

**pour** i = 1 **à** n **faire**

$\quad$ x(i) = y(i)

**finpour**

`x = D\(b – (L + U)x)`

*Gauss-Siedel*

**pour** i = 1 **à** n **faire**

$\quad$ x(i) = $\dfrac{b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^{n} A_{ij}x_j}{A_{ii}}$

**finpour**

`x = (D + L)\(b – Ux)`

*Relaxation*

**pour** i = 1 **à** n **faire**

$\quad$ x(i) = $\omega\,\dfrac{b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^{n} A_{ij}x_j}{A_{ii}} + (1-\omega)x_i$

**finpour**

`x = (D + ωL)\((1-ω)D – ωU)x + ωb)`

**fintantque**

### Notations matricielle des méthodes

$$Mx^{(k+1)} = Nx^{(k)} + b \Leftrightarrow x^{(k+1)} = \underbrace{M^{-1}N}_{C}\,x^{(k)} + \underbrace{M^{-1}b}_{d} \Leftrightarrow \boxed{x^{(k+1)} = Cx^{(k)} + d}$$

## II. Condition suffisante de convergence

$$e^{(k+1)} = x^{(k+1)} - \tilde{x} \underset{\substack{\tilde{x}=C\tilde{x}+d \\ \text{par def}}}{=} C\left(x^{(k)} - \tilde{x}\right) = C^2\left(x^{(k-1)} - \tilde{x}\right) = C^{k+1}(x_0 - \tilde{x})$$

$$\left\|e^{(k+1)}\right\| = \left\|C^{k+1}(x_0 - \tilde{x})x_0 - \tilde{x}\right\| \le \|x_0 - \tilde{x}\| \underbrace{\|\mathrm{C}\|^{k+1}}_{\to 0\,?}$$

- La suite de vecteurs converge si il existe une norme telle que $\|\mathrm{C}\| < 1$.
- Si $A$ est à diagonale dominante, Jacobi et Gauss-Seidel convergent
- Si $A$ est symétrique définie positive, la relaxation converge pour $0 < \omega < 2$

## III. Conditionnement et stabilité

$$A(x + \delta_x) = (b + \delta_b) \;\Rightarrow\; A\delta_x = \delta_b \;\Rightarrow\; \delta_b = A^{-1}\delta_x$$

$$\frac{\|\delta_x\|}{\|x\|} \le \underbrace{\|A^{-1}\|\|A\|}_{cond(A)} \frac{\|\delta_b\|}{\|b\|} \qquad cond(A) = \frac{|\lambda_1|}{|\lambda_n|}$$ bon si proche de 1, mauvais si $\gg 1$

## IV. Méthode du gradient

$$Ax = b \Leftrightarrow \min_x \frac{1}{2}x^TAx - b^Tx \qquad x^{(k+1)} = x^{(k)} + \rho d \qquad \underbrace{d = -(Ax - b)}_{\text{direction de descente}} \qquad \underbrace{\rho = -\frac{d^T\left(Ax^{(k)} - b\right)}{d^TAd}}_{\text{pas de descente}}$$

# Outils mathématiques

$A$ a diag. dominante $\Leftrightarrow \quad \forall i \;\; |A_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} A_{ij}$

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

## I. Normes vectorielles

$$\boxed{\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}}$$

$f(x)$ norme $\Leftrightarrow f(x) \ge 0$ et $\forall x$ et $f(x) = 0 \; si \; x = 0$

$f(\lambda x) = |\lambda| f(x)$

$f(x + y) \le f(x) + f(y)$

$\|x^Ty\|_p \le \|x\|_p\|y\|_q$ avec $\dfrac{1}{1} + \dfrac{1}{q} = 1$

## II. Norme matricielles de type vectorielle : Frobenius

$$\boxed{\|A\|_F^2 = \sum_i \sum_j C_{ij}^2}$$

Pas sous-multiplicative ($\|AB\| \nleq \|A\|\|B\|$)

## III. Norme matricielle d'opérateur

$$\boxed{\|A\| = \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|}}$$

$\|A\|_1 = \max_j \sum_i |A_{ij}|$

$\|A\|_\infty = \max_i \sum_j |A_{ij}|$

$\|A\|_2 = \max_i \sqrt{\mu_i} \le \sqrt{\|A\|_1\|A\|_\infty}$

$\|kA\| = |k|\|A\|$

$\|A + B\| \le \|A\| + \|B\|$

$\|AB\| \le \|A\|\|B\|$

# Rappels

## I. Factorisation *LU*

$$\underbrace{M_{n-1}\dots M_1}_{M=L^{-1}} A = U \qquad \tau_k^T = \Big(\underbrace{0,\dots,0}_{k \text{ fois}}, \frac{a_{k+1,k}}{a_{kk}}, \dots, \frac{a_{nk}}{a_{kk}}\Big) \qquad M_k = I - \tau_k e_k^T \qquad L_k = I + \tau_k e_k^T \qquad L_kL_{k'} = I + \tau_k e_k^T + \tau_{k'} e_{k'}^T \;\; k < k'$$

## II. Factorisation *LDM*

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

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

$$LDL^T \;:\; \boxed{\underset{1\le i\le j-1}{v(i)} = D(i,i)L(j,i)} \qquad \boxed{v(j) = A(j,j) - L(j,1{:}j-1)v(1{:}j-1)}$$

## III. Factorisation de Cholesky : $A = GG^T$

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

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

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