

Factorisation QR

I. Factorisation QR

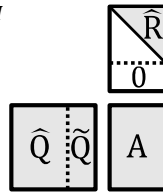
$$A = QR = \hat{Q}\hat{R} \quad \begin{matrix} Q \text{ orthogonale} \\ R \text{ tri. sup.} \end{matrix} \quad Q^T Q = Q Q^T = \hat{Q}^T \hat{Q} = I \quad \hat{Q} \hat{Q}^T \neq I$$

$$H_v = I - \frac{2}{\underbrace{v^T v}_{\beta}} v v^T$$

Orthogonale, symétrique, $H_v = K_{\gamma v}$ ($\gamma \in \mathbb{R}^*$)

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

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



Algorithmes

a. Calcul du vecteur de House

```

fonction [v, β] = vecteurDeHouse(x)
norm_x = x'*x
v = x / sqrt(norm_x)
v(1) = v(1) + sqrt(norm_x)
β = 2 * v(1)^2 / (norm_x + v(1)^2 - x(1)^2)
v = v / v(1)
    
```

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
    
```

b. Produit House-vecteur

```

fonction x = House_vecteur(x, v, β)
w = v' * x
x = x - β * w * 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 $Bx = d$

$$\min_x \underbrace{\|Bx - d\|^2}_{J(x)} = x^T A x - 2x^T b \Leftrightarrow \boxed{\nabla_x J(x) = 2Ax - 2b = 0} \Leftrightarrow \boxed{Ax = b} \quad A = B^T B \quad b = B^T d$$

$$\text{Formules de gradient : } \nabla_x (x^T a) = a \quad \nabla_x (x^T A x) = Ax + A^T x$$

Utilisation de QR

$$B = QR \quad A = R^T R \Leftrightarrow Ax - b = \hat{R}^T (\hat{R}x - \hat{Q}^T d) = 0 \Leftrightarrow \boxed{x = \hat{R}^{-1} \underbrace{\hat{Q}^T d}_c}$$

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

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

$$Av_i = \lambda_i v_i$$

$$AV = VD$$

$$v \rightarrow \lambda : \lambda = \frac{v^T A v}{v^T v} = \frac{v^T A v}{\|v\|^2} = \frac{v^T A v}{\text{si } \|v\|^2=1}$$

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

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

cercles de Gershgorin : contiennent tous les λ

II. Valeurs singulières

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

$$Bv_i = \mu_i u_i \quad B^T u_i = \mu_i v_i$$

$$B = UDV^T$$

$$A = B^T B \Rightarrow (\mu_i^2, v_i) \text{ val. et vect. p. de } A$$

III. Matrice carrée symétrique définie positive

$$(\lambda_i, v_i) \in \mathbb{R} \times \mathbb{R}^p \quad \lambda_1 \geq \lambda_2 \geq \dots \geq 0 \quad v_i^T v_i = 1$$

$$v_i^T v_j = 0 \quad V^T V = I \quad (\text{base de } \mathbb{R}^n)$$

conv°

IV. Méthode de la puissance itérée

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

Déflation : vp suivante

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

tant que (non conv) **faire**
z = Az
z = z / ||z||

fin tant que
λ = z'Az

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

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

V. Calcul de valeurs propres avec QR

$$\begin{cases} Z_k &= A Q_k \\ (Q_{k+1}, R_{k+1}) &= qr(Z_k) \end{cases} \quad T_k = Q_k^T A Q_k \rightarrow D$$

$$T_k = Q_k^T A Q_k = Q_k^T Q_{k+1} R_{k+1} = Q_{k+1}^{(2)} R_{k+1} \Rightarrow \text{On a une suite de } T_k \text{ calculable via QR}$$

$$T_{k+1} = Q_{k+1}^T A Q_{k+1} = R_{k+1} Q_{k+1}^{(2)}$$

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

fin tant que

fonction [Z, D] = Ite_QR_tridiag(A)

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

Tridiagonalisation

fonction A = tri_diag (A)

pour k = 1 **à** n-2 **faire**

[v, β] = vecteurDeHouse(A(k+1:n,k))
p = β * A(k+1:n,k+1:n) * v

w = p - (β * v' * p / 2) * v

A(k+1,k) = A(k,k+1) = ||A(k+1:n,k)||

A(k+1:n,k+1:n) = A(k+1:n,k+1:n) - v*w' - w*v'

finpour

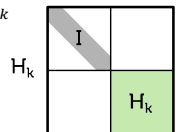
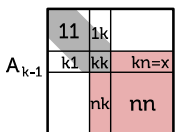
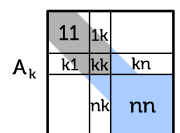
% x = A_{kn}

% p = β A_{nn} v

% w = p - $\frac{\beta v^T p}{2}$ v

% A_{nk}^{(k)} = A_{kn}^{(k)T} = -\alpha e_1}

% A_{nn}^{(k)} = H_k^T A_{nn} H_k = A_{nn} - v w^T - w v^T}}}



Méthodes itératives

I. Les méthodes itératives

Cherche solution de $Ax = b$ avec $(x^{(k)})_{k \in \mathbb{N}} \rightarrow \tilde{x}$ $Ax = b \Leftrightarrow \sum_{j=1}^{i-1} A_{ij}x_j + A_{ii}x_i + \sum_{j=i+1}^n A_{ij}x_j = b_i$

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

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

Jacobi	pour i = 1 à n faire	$y(i) = \frac{b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^n A_{ij}x_j}{A_{ii}}$	Forme matricielle (A = D + U + L)
	finpour		
	pour i = 1 à n faire	$x(i) = y(i)$	$x = D \setminus (b - (L + U)x)$
	finpour		
Gauss-Siedel	pour i = 1 à n faire	$x(i) = \frac{b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^n A_{ij}x_j}{A_{ii}}$	$x = (D + L) \setminus (b - Ux)$
	finpour		
Relaxation	pour i = 1 à n faire	$x(i) = \omega \frac{b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^n A_{ij}x_j}{A_{ii}} + (1 - \omega)x_i$	$x = (D + \omega L) \setminus ((1 - \omega)D - \omega U)x + \omega b$
	finpour		
	fintantque		

Notations matricielle des méthodes

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

II. Condition suffisante de convergence

$$e^{(k+1)} = x^{(k+1)} - \tilde{x} = C(x^{(k)} - \tilde{x}) = C^2(x^{(k-1)} - \tilde{x}) = C^{k+1}(x_0 - \tilde{x})$$

$\tilde{x} = C\tilde{x} + d$
par def

$$\|e^{(k+1)}\| = \|C^{k+1}(x_0 - \tilde{x})\| \leq \|x_0 - \tilde{x}\| \underbrace{\|C\|^{k+1}}_{\rightarrow 0?}$$

- La suite de vecteurs converge si il existe une norme telle que $\|C\| < 1$.
- 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

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

$$\frac{\|\delta_x\|}{\|x\|} \leq \frac{\|A^{-1}\| \|A\|}{\text{cond}(A)} \frac{\|\delta_b\|}{\|b\|} \quad \text{cond}(A) = \frac{|\lambda_1|}{|\lambda_n|} \text{ bon si proche de 1, mauvais si } \gg 1$$

IV. Méthode du gradient

$$Ax = b \Leftrightarrow \min_x \frac{1}{2} x^T A x - b^T x \quad x^{(k+1)} = x^{(k)} + \rho d \quad d = -(Ax - b) \quad \rho = -\frac{d^T (Ax^{(k)} - b)}{d^T A d}$$

direction de descente pas de descente

Outils mathématiques

$$A \text{ a diag. dominante} \Leftrightarrow \forall i \quad |A_{ii}| > \sum_{j \neq i}^n |A_{ij}| \quad A \text{ définie positive} \Leftrightarrow \begin{cases} A \text{ symétrique} \\ \forall x \neq 0, x^T A x > 0 \end{cases}$$

I. Normes vectorielles

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

$$\begin{aligned} f(x) \text{ norme} &\Leftrightarrow f(x) \geq 0 \text{ et } \forall x \text{ et } f(x) = 0 \text{ si } x = 0 \\ f(\lambda x) &= |\lambda| f(x) \\ f(x + y) &\leq f(x) + f(y) \\ \|x^T y\|_p &\leq \|x\|_p \|y\|_q \text{ avec } \frac{1}{p} + \frac{1}{q} = 1 \end{aligned}$$

II. Norme matricielle de type vectorielle : Frobenius

$$\|A\|_F^2 = \sum_i \sum_j C_{ij}^2 \quad \text{Pas sous-multiplicative } (\|AB\| \not\leq \|A\| \|B\|)$$

III. Norme matricielle d'opérateur

$$\begin{aligned} \|A\|_1 &= \max_j \sum_i |A_{ij}| \\ \|A\|_\infty &= \max_i \sum_j |A_{ij}| \\ \|A\|_2 &= \max_i \sqrt{\mu_i} \leq \sqrt{\|A\|_1 \|A\|_\infty} \end{aligned}$$

$$\|kA\| = |k| \|A\| \quad \|A + B\| \leq \|A\| + \|B\| \quad \|AB\| \leq \|A\| \|B\|$$

Rappels

I. Factorisation LU

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

II. Factorisation LDM

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

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

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

III. Factorisation de Cholesky : A = GG^T

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

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

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

