Méthodes itératives classiques

Jacobi

$$\begin{cases} a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n = b_1 \\ a_{21} x_1 + a_{22} x_2 + a_{23} x_3 + \cdots + a_{2n} x_n = b_2 \\ \vdots \\ a_{n1} x_1 + \cdots + a_{nn} x_n = b_n \end{cases}$$

$$x_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - a_{12} x_2^{(k)} - \cdots - a_{1n} x_n^{(k)} \right)$$

$$x_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - a_{21} x_1^{(k)} - \cdots - a_{2n} x_n^{(k)} \right)$$

$$x_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1} x_1^{(k)} - \cdots - a_{nn-1} x_{n-1}^{(k)} \right)$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

⚠ PyMaO, 2 vecteurs

Mieux (?) GS : utiliser dernier itéré disponible

$$\begin{cases} x_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - a_{21} x_1^{(k+1)} - a_{23} x_3^{(k)} - \cdots - a_{2n} x_n^{(k)} \right) \\ \vdots \\ x_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1} x_1^{(k+1)} - \cdots - a_{nn-1} x_{n-1}^{(k+1)} \right) \end{cases}$$

Ordre compte.

Description matricielle

$D = \text{diag}\, A$

$L = \text{lower}(A)$, $U = \text{upper}(A)$

$$A = L + D + U$$

Jacobi : $x^{k+1} = -D^{-1}\left((L+U) x^k + b\right)$ $\qquad M_J = -D^{-1}(L+U)$

Gauss-Seidel $\quad (D + L) x^{k+1} = b - U x^k$ $\qquad M_{GS} = -(D+L)^{-1} U$

$M_J = D$, $\quad N_J = -(L+U)$

$M_{GS} = D+L \quad N_J = -U$

General "Matrix splitting"

$$M x^{k+1} = N x^k + b, \qquad M \text{ inversible}$$

$A = M - N$

Convergence : rayon spectral $\rho(C) = \max\{|\lambda|, \lambda \text{ vp de } C\}$

Th (Ciarlet, Th 5.1.1)
Méthode itérative cv $\Leftrightarrow$ $\rho(M^{-1}N) < 1$ [$\rho(B) \leq \|B\|$ pour norme subordonnée]

Matrice d'itération $x^{k+1} = M^{-1}N x^k + M^{-1}b$

Rq $M^{-1}N = I - M^{-1}A$

Exemples pour Jacobi ou GS

Th A diagonale strictement dominante $\Rightarrow$ Jacobi cv
DSD : $|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \forall i$ ($\Rightarrow$ inversible)

Th A SPD $\Rightarrow$ GS converge

Sur relaxation

$$x^{k+1} = \omega\, x_{GS}^{(k+1)} + (1-\omega)\, x^k \qquad \omega \in \mathbb{R}$$

$$x^{k+1} = \omega\left(b - (D+L)^{-1} U x^k\right) + (1-\omega)\, x^k$$

$$\Leftrightarrow (D+L)\, x^{k+1} = \omega b - \omega U x^k + (1-\omega)(D+L)\, x^k$$

$$x^{k+1} = \omega (D+L)^{-1}(b - U x^k) + (1-\omega)\, x^k$$

$$(D+L)\, x^{k+1} = \omega b - \omega U x^k + (1-\omega)(D+L)\, x^k$$

Sur relaxation

$$x^{k+1} = \omega x_{GS}^{(k+1)} + (1-\omega) x^k$$

$$x^{k+1} = \omega (D+L)^{-1} (b - U x^k) + (1-\omega) x^k$$

$$D x^{k+1} - D x^k = \omega (b - U x^k - L x^{k+1}) - \omega D x^k$$

$$(D + \omega L) x^{k+1} = \omega b + ((1-\omega) D - \omega U) x^k$$

$$M_\omega = (D+\omega L)^{-1}, \quad N_\omega = (1-\omega) D - \omega U$$

(Th) Alors CV $\Rightarrow$ $0 < \omega < 2$

c) A sym et déf pos, $0 < \omega < 2 \Rightarrow$ son CV.

Méthode par blocs $A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ \vdots & & & \\ A_{n1} & & & A_{nn} \end{bmatrix}$

A_{ii} inversible

Jacobi / bloc

$$A_{ii} x_i^{k+1} = b_i - \sum_{j \neq i} A_{ij} x_j \quad \text{bloc}$$

Gs / bloc $A_{ii} x_i^{k+1} = b_i - \sum_{j<i} A_{ij} x_j^{k+1} - \sum A_{ij} x_j^k$

Méthode du gradient conjugué (1)

Notations - généralités

$Ax = b$, $x \in \mathbb{R}^d$, $b \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$ inversible

A SPD $\quad A^T = A$, $\forall y \; y^T A x = x^T A y$

Solution x^* $\quad x^* = A^{-1} b$

A "grande et creuse" cf exemple suite SPARSE

Suite $(x_k)_{k \in \mathbb{N}}$ $\quad x_k \longrightarrow x^*$?

Résidu $\quad r = b - Ax$, $\quad r_k = b - A x_k$

Erreur $\quad e = x - x^*$

$r = -Ae$

Lemme. $\quad \dfrac{\|e\|}{\|e_0\|} \leq \kappa(A) \dfrac{\|r\|}{\|r_0\|}$ $\qquad \kappa(A) = \|A\| \, \|A^{-1}\|$

Dem

$r = b - Ax = -Ae$

$\|e\| = \|A^{-1} A e\| \leq \|A^{-1}\| \, \|r\|$

$\|r_0\| = \|A e_0\| \leq \|A\| \, \|e_0\|$

$\dfrac{\|e\|}{\|e_0\|} \leq \dfrac{\|A^{-1}\| \, \|r\| \, \|A\|}{\|e_0\| \, \|A\|} = \kappa(A) \dfrac{\|r\|}{\|r_0\|}$

⚠ Critère d'arrêt. (+ tard)

Sys linéaire $\leftrightarrow$ minimisation et $\perp$

$J(x) = \frac{1}{2} x^T A x - b^T x$

Th 1) $\min_x J(x) \Leftrightarrow Ax = b$ si A SPD, $\exists!$ min

2) $F \subset \mathbb{R}^d$, $\min_{x \in F} J(x) \Leftrightarrow r \perp F$, $\exists! \, x \in F$

Plan sur CG

- Generalités notations
- Lien minimisation - orthogonalité
- Présentation "abstraite" → algorithme
- Discussion algo / pratique (matrix free)
- Critère d'arrêt
- Convergence
- Compléments ?

Minimisation orthogonalité et normes — A SPD

$$J(x) = \frac{1}{2} x^T A x - x^T b$$

$\min_{x \in \mathbb{R}^d} J(x) \iff Ax = b$ (gradient, ou à la main)

Si A SPD, $\|x\|_A = (x^T A x)^{1/2}$ norme $(x, y)_A = y^T A x$ pdt scalaire

(Rq) $\|x - x^*\|_A^2 = 2 J(x) + x^* A x$

Dem $\|x - x^*\|_A^2 = (x - x^*)^T A (x - x^*) = x^T A x - \underbrace{2 x^T A x^*}_{\text{Asym}} + x^* A x^*$

$$= \underbrace{x^T A x - 2 x^T b}_{2J(x)} + \underbrace{x^* A x^*}_{ck}$$

donc $\min_{x \in S} J(x) \iff \min_{x \in S} \|x - x^*\|_A \quad S \subset \mathbb{R}^d$

$\iff$ projection de x^* sur S (convexe !)

Si F = ssev de $\mathbb{R}^d$ $\min_{x \in x_0 + F} \|x - x^*\|_A \iff \begin{cases} x \in x_0 + F \\ x - x^* \perp_A F \end{cases}$

$x - x^* \perp_A F \iff \forall y \in F,\ y^T A (x - x^*) = y^T A x - y^T b = 0$

$\iff \underline{r \perp F}$ ($\perp$ pas A) !!

Ssespace Krylov $K_m(A, r_0) = K_m(r_0) = \text{vect}(r_0, A r_0, \dots, A^{m-1} r_0)$

- $K_m(r_0) \subset K_{m+1}(r_0)$,
- $\dim K_m(r_0) \leq m$

$x \in K_m(r_0) \iff x = p(A) r_0$, $p \in \mathbb{R}[X]$, $\deg p \leq m-1$

Si $\begin{cases} K_m(r_0) = K_{m+1}(r_0) \Rightarrow A^m r_0 \in K_m(r_0) \\ A^m r_0 = r_0 + A q(A) r_0 \quad \deg q < m-1 \\ b = A x_0 + A \underbrace{(A^{m-1} - q(A))}_{\neq 0} r_0 \Rightarrow x_0 + (A^{m-1} - q(A)) r_0 \text{ sol}^0 \end{cases}$

De toutes façons $\text{sol}^0 \in K_d(r_0)$ (!)

Algo du gradient conjugué

x_k realise le min de $J(x)$ sur $x_0 + K_k(r_0)$

(Bien défini, th projection sur cvxe)

(1) $r_k \in K_k$ si $l < k$: $r_l = b - A x_l = r_0 + A(x_0 - x_l)$
$\in K_{l+1} + A K_l \subset K_{l+1} \subset K_k$

Récurrence sur l
$r_0 \in K_k$ (déf)
r_{l+1} $l \Rightarrow l+1$: $r_{l+1} = b - \dots$

(2) $r_k^* r_l = 0$ $l < k$. $r_k \perp K_k$ et $r_k \in K_k$ OK

Donc (?) $K_k = \text{vect}(r_0, \dots, r_{k-1})$ (⊃, clm dimension)
↳ tant que $r_k \neq 0$ (mais si $r_k = 0$, x_k est solu)

(3) Rq si $x_k = x_{k+1}$ alors $r_k = r_{k+1}$. Mais (2) ⇒
$\|r_k\|^2 = r_k^T r_k = r_{k+1}^* r_k = 0 \Rightarrow x_k = x^*$

(4) On définit p_{k+1} par $x_{k+1} = x_k + \alpha_k p_{k+1}$ (α_k pas défini mais $\alpha_k \neq 0$)

p_{k+1} déterminé à un scalaire près par :
$p_{k+1} \in K_{k+1}$; $y^T A p_{k+1} = 0$, $\forall y \in K_k$

p_k est $A\perp$ à K_k (on dit conjugué ...)

Dem : $r_{k+1} = b - A x_k - \alpha_k A p_{k+1} = r_k - \alpha_k A p_{k+1}$ $\perp K_{k+1}$
En particulier, $\forall y \in K_k$ $r_k \perp y \Rightarrow A p_{k+1} \perp y$
$p_{k+1} \in K_{k+1}$ ($= x_{k+1} - x_k$), et $\perp_A$ à l'ssev de dim k de rang

(5) Ñ raisonn' $\text{vect}(p_1, \dots, p_k) = K_k$ tant que $p_k \neq 0 \Leftrightarrow r_k \neq 0$

④ On définit p_k par :

$$p_0 = 0, \qquad x_{k+1} = x_k + \alpha_k p_k$$

(α_k pas défini, $\neq 0$)

p_k déterminé à un scalaire près par

$$p_k \in K_{k+1}, \quad y^T A p_k = 0, \ \forall y \in K_k$$

→ p_k est A⊥ à K_k (on dit conjugué)

Dem
$r_{k+1} = b - A x_k - \alpha_k A p_k = r_k - \alpha_k A p_k$
$r_{k+1} \perp K_{k+1}$, $y \in K_k \Rightarrow r_k \perp y \Rightarrow A p_k \perp y$
$\Rightarrow p_k \in K_{k+1}$

$p_k \in K_{k+1}$ ($\parallel x_{k+1} - x_k$) et $\perp_A$ à 1 sev de dim 1 de moins

⑤ $\hat{\text{M}}$ raisonnement $\text{vect}(p_0, \ldots, p_k) = K_k$ tant que $p_k \neq 0 \Leftrightarrow r_k \neq 0$

⑥ Déterminer $p_k = \underbrace{r_k}_{\substack{K_{k+1} \\ \perp K_k}} + \underbrace{\sum_{j=0}^{k-1} \gamma_{kj} p_j}_{K_k}$

Rq $r_k^T A p_k \neq 0$

Dem déf $p_k \Longrightarrow$ $r_{k+1} = r_k - \alpha_k A p_k$

$$\underbrace{r_k^T r_{k+1}}_{= 0} = \underbrace{\|r_k\|^2}_{\neq 0} - \alpha_k r_k^T A p_k \Longrightarrow \neq 0$$

MQ
$\gamma_{kj} = 0, \ j = 0, \ldots, k-2$
$\gamma_{k k-1} = \beta_{k-1} = -\dfrac{r_{k-1}^T A r_k}{r_{k-1}^T A p_{k-1}}$

On utilise ④ sous la forme $r_l^T A p_k = 0, \ \forall l = 0, \ldots, k-1$

Pour $l = 0, \dots k-2$: $r_l \perp K_k$, $A r_l \in K_{k+1} \Rightarrow A r_l \perp r_k$

Dans la somme $r_l^T A p_j = 0$ pour $j \neq l$

et $r_l^T A p_l \neq 0 \Rightarrow \gamma_{kl} = 0, \ l = 0, \dots k-2$

③ $p_k = r_k + \beta_{k-1} p_{k-1}$

Pdt scalaire avec $A r_{k-1}$ ($r_{k-1}^T A p_{k-1} \neq 0$

$$\boxed{\beta_{k-1} = - \frac{r_k^T A r_{k-1}}{p_{k-1}^T A r_{k-1}}}$$

④ Calcul de α_k ? α_k minimise J sur la direction p_k

$g(\alpha) = J(x_k + \alpha p_k)$

$g'(\alpha) = 0 \Rightarrow$

$g(\alpha) = \frac{1}{2} \alpha^2 p_k^T A p_k + \alpha p_k^T A x_k + \frac{1}{2} x_k^T A x_k - x_k^T b - \alpha p_k^T b$

$g'(\alpha) = \alpha p_k^T A p_k + p_k^T A x_k - p_k^T b = \alpha p_k^T A p_k - p_k^T r_k$

$$\boxed{\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}}$$

⑤ Meilleures formules pour α_k, β_k

$$\boxed{\alpha_k = \frac{\|r_k\|^2}{p_k^T A p_k} \qquad \beta_k = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}}$$

Dem

1) $p_k = r_k + \beta_{k-1} p_{k-1}$ + $p_{k-1} \in K_k$, $r_k \perp K_k$

$r_k^T p_k = \|r_k\|^2 + \beta_{k-1} \underbrace{r_k^T p_{k-1}}_{=0} \Rightarrow r_k^T p_k = \|r_k\|^2$

2) $r_{k+1} = r_k - \alpha_k A p_k$

$0 = p_k^T r_{k+1} = p_k^T r_k - \alpha_k p_k^T A p_k \Rightarrow \alpha_k = \frac{p_k^T r_k}{p_k^T A p_k} = \frac{\|r_k\|^2}{p_k^T A p_k}$

(3) $p_{k+1} = r_{k+1} + \beta_k p_k$

$p_{k+1}^T A p_k = r_{k+1}^T A p_k + \beta_k p_k^T A p_k$

$= 0$ (conjugué) $\Rightarrow \beta_k = - \dfrac{r_{k+1}^T A p_k}{p_k^T A p_k}$

puis $p_k^T A p_k = p_k^T A (r_k + \beta_{k-1} p_{k-1})$

$= p_k^T A r_k + \beta_{k-1} p_k^T A p_{k-1}$

et (rappel (6)) : $p_k^T A r_k = \dfrac{\|r_k\|^2}{\alpha_k}$

Ensemble : $\beta_k = - \dfrac{r_{k+1}^T A p_k}{\|r_k\|^2} \alpha_k$

Enfin $r_{k+1} = r_k - \alpha_k A p_k$

$\|r_{k+1}\|^2 = - \alpha_k r_{k+1}^T A p_k$

$\Rightarrow \beta_k = \dfrac{\|r_{k+1}\|^2}{\|r_k\|^2}$.

Algo cf Wikipedia !

Données : x_0, itmax, tol.

$r_0 = b - A x_0$, $p_0 = r_0$; ~~k=0~~

Pour $k = 0, k+1, k \leq$ itmax

$\alpha_k = \dfrac{\|r_k\|^2}{p_k^T A p_k}$

$x_{k+1} = x_k + \alpha_k p_k$

$r_{k+1} = r_k - \alpha_k A p_k$

Si $\|r_{k+1}\| < tol \, \|b\|$ STOP ← Arrêt

$\beta_{k+1} = \dfrac{\|r_{k+1}\|^2}{\|r_k\|^2}$

$p_{k+1} = r_{k+1} + \beta_k p_k$

e) $r_k = r_{k-1} - \alpha_k A p_k \Rightarrow \|r_k\|^2 = -\alpha_k r_k^T A p_k$

$\rightarrow \boxed{\beta_{GM} = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}}$

Osg !!!

Algo final (cf Allaire / Kaber)

Init $x \in \mathbb{R}^d$. $r = b - Ax$, $p = r$, $\gamma = \|r\|^2$

Tantque pas convergé faire

$y = Ap$

$\alpha = \gamma / p^T y$

$x = x + \alpha p$

$r = r - \alpha y$

$\beta = \|r\|^2 / \gamma$

$\gamma = \|r\|^2$

$p = r + \beta p$

Mémoire 3 vecteurs x, r, p.

coût/iter 3 axpys, 2 pdt scalaires, 1 matvec

↳ Implémentation : Pas besoin de A, juste $p \to Ap$

"Convergence" en au plus n iters (mais pas intéressant en FP)

Test d'arrêt Réduction du résidu, donc quand γ petit

En pratique souvent $\frac{\gamma}{\|r_0\|} \leq \varepsilon$ FP0

mais si r_0 petit trop difficile $\Rightarrow$

Mieux $\gamma \leq \varepsilon \|b\|$ (souvent parfait)

Convergence de CG

On repart de la def de x_k

$x_k \in x_0 + K_k$, minimise $\|x - x^*\|_A$ sur $x_0 + K_k$

Deja dit

$$x \in x_0 + K_k \Rightarrow x - x^* = x_0 - x^* + \sum_{j=0}^{k-1} \gamma_j A^j r_0$$

$$r_0 = A x^* - A x_0$$

$$x - x^* = p(A)(x^* - x_0)$$

avec $p(z) = 1 - \sum_{j=0}^{k-1} \gamma_j z^{j+1}$, $\deg p \le k$, $p(0) = 1$

$$\|x_k - x^*\|_A = \min_{\substack{p \in P_k \\ p(0)=1}} \|p(A)(x^* - x_0)\|_A$$

A SPD. $A = U \Lambda U^T$

U matrice orthogonale

Λ diagonale $\Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix}$

$p(A) = U p(\Lambda) U^T$ $\quad p(A) = \mathrm{diag}(p(\lambda_1), \dots, p(\lambda_d))$

On note $A^{1/2} = U \Lambda^{1/2} U^T$, $\|x\|_A^2 = x^T A x = \|A^{1/2} x\|_2^2$

Donc $\|p(A) e\|_A^2 = \|A^{1/2} p(A) e\|_2^2 \le \|p(A)\|_2^2 \, \|A^{1/2} x\|_2^2$

$= \|p(A)\|_2^2 \cdot \|x\|_A^2$

$$\|p(A)\|_2 = \max_{\lambda \in \sigma(A)} |p(\lambda)|$$

Thm $\quad \|x_k - x^*\| \le \left(\min_{\substack{p \in P_k \\ p(0)=0}} \max_{\lambda \in \sigma(A)} |p(\lambda)| \right) \|x_0 - x^*\|_A$

Corollaire $\quad p \in P_k$, $p(0) = 1$.

$$\|x_k - x^*\| \le \max_{\lambda \in \sigma(A)} |p(\lambda)| \, \|x_0 - x^*\|_A$$

Bien choisir le polynôme.

On retrouve conv en d itérations

$\nexists$ Prendre $p(\lambda) = \prod_{i=1}^{d}\left(1 - \frac{\lambda}{\lambda_i}\right)$ 0 sur le spect

Corollaire. Estimation "classique"

$$\max_{\lambda \in \sigma(A)} |p(\lambda)| \leq \max_{\lambda_1 \leq \lambda \leq \lambda_d} |p(\lambda)|$$

et le pb

$$\left| \min_{\substack{p \in \mathcal{P}_k \\ p(0)=1}} \max_{\lambda \in [\lambda_1, \lambda_d]} |p(\lambda)| \right|$$

a une sol° connue (polynomes de Tchebyteleff)

$$p(\lambda) = \frac{T_k\left(\frac{\lambda_1 + \lambda_d - 2\lambda}{\lambda_d - \lambda_1}\right)}{T_k\left(\frac{\lambda_1 + \lambda_d}{\lambda_d - \lambda_1}\right)}$$

$T_k(t) = \cos(k \arccos(t))$ $\quad -1 \leq t \leq 1$

Majoration erreur

$$\frac{\|x_k - x^*\|}{\|x_0 - x^*\|} \leq \frac{1}{T_k\left(\frac{\lambda_1 + \lambda_d}{\lambda_d - \lambda_1}\right)} \leq 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k$$

$\kappa(A) = \frac{\lambda_d}{\lambda_1} = \text{cond}_2(A)$

Si $\kappa \gg 1$ réduction de $\varepsilon_k =$

$$k \simeq \frac{\sqrt{\kappa(A)}}{2} \ln \frac{2}{\varepsilon} \quad \text{en } \sqrt{\kappa(A)}$$

$|T_k(t)| \leq 1, \ |t| \leq 1$; $|C_k(t)| = \frac{1}{2}\left(t + \sqrt{t^2-1}\right)^k + \frac{1}{2}\left(t - \sqrt{t^2-1}\right)^k \quad t \geq 1$

$|C_k(t)| \geq \frac{1}{2}\left(t + \sqrt{t^2-1}\right)^k$

Preconditionnement :

Remplacer $Ax = b$ par $\underline{M^{-1}Ax = M^{-1}b}$

① $M^{-1}A$ "mieux conditionné" $\Rightarrow M \simeq A$

② M "facile à inverser" $\Leftrightarrow$ compromis

⚠ $M^{-1}A$ pas symétrique

Si $M = SDP \Rightarrow M = LL^T$ et on remplace par

$$\hat{A}\hat{x} = \underbrace{L^{-1}AL^{-T}}_{SDP}\hat{x} = L^{-1}b = \hat{b}$$

mais on peut travailler avec A, M directement

On définit $\hat{x}_k = L^T x_k$; $\hat{p}_k = L^T p_k$,

Pour $\hat{r}_k = \hat{b} - \hat{A}\hat{x}_k = L^{-1}b - L^{-1}AL^{-T}\hat{x}_k$

$$L\hat{r}_k = b - Ax_k = r_k \Rightarrow \hat{r}_k = L^{-1}r_k,$$

[Pratique $\hat{r}_k = L^T z_k$, et $L^{-1}r_k = L^T z_k \Rightarrow \underline{Mz_k = r_k}$ (solve)]

Algo :

$$\hat{\alpha}_k = \frac{\hat{r}_k^T \hat{r}_k}{\hat{p}_k^T \hat{A} \hat{p}_k} = \frac{z_k^T L L^T z_k}{\hat{p}_k^T L^{-1} A L^{-T} \hat{p}_k} = \frac{(Mz_k)^T z_k}{p_k^T A p_k} = \frac{z_k^T r_k}{p_k^T A p_k}$$

$$\hat{x}_{k+1} = x_k + \hat{\alpha}_k \hat{p}_k \Rightarrow x_{k+1} = x_k + \hat{\alpha}_k p_k$$

$$\hat{r}_{k+1} = \hat{r}_k - \hat{\alpha}_k L^{-1} A L^{-T} \hat{p}_k \Rightarrow L\hat{r}_{k+1} = r_k - \hat{\alpha}_k A p_k$$

$$\hat{\beta}_k = \frac{\hat{r}_{k+1}^T \hat{r}_{k+1}}{\hat{r}_k^T r_k} = \frac{r_{k+1}^T L^{-T} L^{-1} r_{k+1}}{r_k^T L^{-T} L^{-1} r_k} = \frac{z_{k+1}^T r_{k+1}}{z_k^T r_k}$$

$$\hat{p}_{k+1} = \hat{r}_k + \hat{\beta}_{k+1} \hat{p}_k \Rightarrow p_{k+1} = \underbrace{L^{-T}\hat{r}_k}_{z_k} + \hat{\beta}_k p_k$$

Comme sans précond, avec $\underline{Mz_k = r_k}$;

résoudre 1 prec. / étape

Rq Aussi avec nouveau pdt scalaire $(x, y)_M = y^T M x$

$M^{-1}A$ est sym $y_0^T M^{-1} A x = y^T A x$ sym

Exemples de preconditionneurs

"simplifications" de A
- coeff constants, ordre + bas, — —
  → solveurs rapides
- Cholesky / LU incomplet / polynôme
- Méthodes itératives (Jacobi / GS, bloc)
- Multigrille / DD

① le + simple $M = \mathrm{diag}(A) = D_A$ $\quad Mz = r \Leftrightarrow z_i = \frac{r_i}{a_{ii}}$

② SSOR Pourquoi ? $\quad A = M - N \quad Ax = b$

$$Mx^{k+1} = Nx^k + b \Rightarrow Mx = Nx + b$$

$$x - M^{-1}Nx = M^{-1}b \qquad A = M - N,\ M^{-1}N = I - M^{-1}A$$

$$\boxed{M^{-1}Ax = M^{-1}b} \Rightarrow \text{méthode itérative}$$

SSOR $\quad A = D - E - E^T \quad E$ = strict sous-diagonal

$$M = (D - \omega E)\, D^{-1}\, (D - \omega E^T) \qquad 0 < \omega < 2$$

Pas de calcul en +.

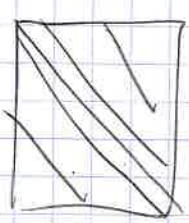
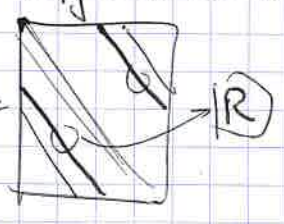
$Mz = r \Leftrightarrow (D - \omega E)y = r$; puis $(D - \omega E^T)z = Dy$

② Factorisation incomplète (ICC(0))

si $A = L_A L_A^T$ "exact" "remplissage" de L_A = A  → L_A 

L ↓q $\quad \begin{cases} L_{ij} = 0 & \text{si } A_{ij} = 0 \\ A_{ij} = (LL^T)_{ij} & \text{si } A_{ij} \neq 0 \end{cases}$

$A = LL^T + R$ avec $R_{ij} = 0$ si $A_{ij} = 0$

Pas A =  $\Rightarrow L R^T =$  → R

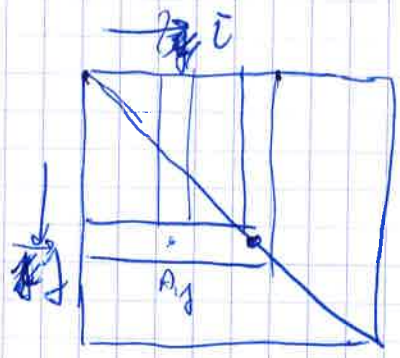
Existe pas toujours (M matrice)

Cmde ichol . Matlab

Algo IC(0) (⚠ structure matrice) $M = LL^T$

$L = A$ (= tril(A))

```
for k = 1 : n
    L_kk = √L_kk
    for i = k+1 : n
        if A_ik ≠ 0
            L_ik = L_ik / L_kk
    for j = k+1 : n
        for i = j : n
            if A_ij ≠ 0
                L_ij = L_ij − L_ik L_jk
```



Ne change pas le conditionnement (mais améliore la répartition des vp) ?

Preconditionnement MIC(0) (Modified)

Assure : $\forall i \quad \sum_{j=1}^{n} M_{ij} = \sum_{j=1}^{n} A_{ij}$ (somme ligne)

$\Rightarrow$ acond de M sur sol° régulière $\simeq$ acond de A

$\Rightarrow$ OK pour sol° régulière, mais non sur sol. irrég

changer l'algo

```
if A_ik ≠ 0
    L_ik = L_ik / L_kk
else
    L_kk = L_kk + L_ik
```

```
if A_ij ≠ 0
    L_ij = L_ij − L_ik L_jk
else
    L_ii = L_ii − L_ik L_jk
```

Cond.' réduit à $O\left(\frac{1}{h}\right)$ $\Rightarrow$ # its $\simeq \sqrt{n}$

DANS CERTAINS CAS

Extensions : Fill-in level | ~~Thre~~ Seuil -- cf STAP

Preconditionnement polynomial

$M^{-1} = S(A)$ Δ polynome "bas degré"

$\Rightarrow$ pas de factorisation //

Exemple : série Neumann

$\omega A = I - (I - \omega A)$ $\quad \omega$ tq $\rho(I - \omega A) < 1$

$(\omega A)^{-1} = \sum_{p=0}^{\infty} (I - \omega A)^p$

$\Rightarrow M_p = \sum_{q=0}^{p} (I - \omega A)^q = I + (I - \omega A) + (I - \omega A)^2 + \dots$

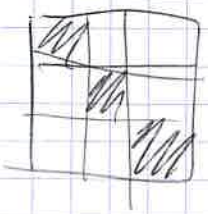
Rem ?

$S \in P_p$, minimiser $\max_{\lambda \in \sigma(A)} |1 - \lambda S(\lambda)|$ $\rightarrow$ trop dur

$\Rightarrow \max_{\lambda \in E} |1 - \lambda S(\lambda)|$, $\quad \underline{\sigma(A) \subset E}$

cf Tchebytcheff (least square)

Bloc Jacobi



$M = \text{bloc}\begin{pmatrix} D_1 & & \\ & \ddots & \\ & & D_N \end{pmatrix}$

// "facile" mais peu efficace

Matrices creuses

① Stockage et calcul avec des matrices creux

② Optimisation en Matlab (1)

↳ ② Parallélisme (distribué)

---

① Exemple

$$A = \begin{pmatrix} 1 & 0 & 0 & 2 & 0 \\ 3 & 4 & 0 & 5 & 0 \\ 6 & 0 & 7 & 8 & 9 \\ 0 & 0 & 10 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{pmatrix}$$

Stockage "coordonnés" ou triplet ou IJ

A	12	9	7	5	1	2	11	3	6	4	8	10
IR	5	3	3	2	1	1	4	2	3	2	3	4
JC	5	5	3	4	1	4	4	1	1	2	4	3

Sans ordre. Mieux.

A	1	2	3	4	5	6	7	8	9	10	11	11
IR	1	1	2	2	2	3	3	3	3	4	4	5
JC	1	4	1	2	4	1	3	4	5	3	4	5

Stockage = $3 \times Nz$

Mieux pointeur vers le début de la ligne (AA, IA, JA)

IA pointeur vers le début des lignes

ligne i : entre $IA(i)$ et $IA(i+1)-1$

Stockage Compressed Sparse Row (morse)

AA	1	2	3	4	5	6	7	8	9	10	11	12
JA	1	4	1	2	4	1	3	4	5	3	4	5
IA	1	3	6	10	12	13						

Variantes CSC (colonne)

DSR sans la diag

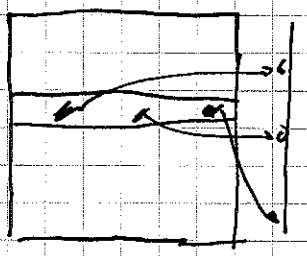
BSR / VBR (variable) block row → matrices bloc

DIAG (pour matrices DF)

Operations :

Matvec $x \to y = Ax$

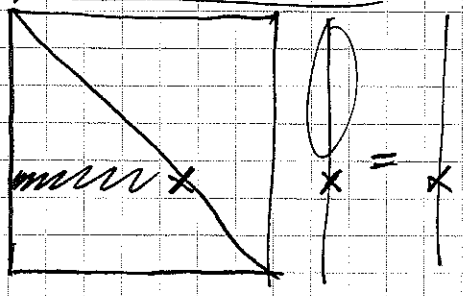
Parcours par ligne.



```
for i = 1 : n
    j1 = IA(i) ; j2 = IA(i+1) - 1 ;
    y_i = Σ_{j=j1}^{j2} A(j) X(JA(j))
```

adressage indirect

Solution système triangulaire



diag = I

$x_1 = b_1$

for i = 2 : n

$j1 = IA(i)$; $j2 = IA(i+1) - 1$

$x_i = \left(b_i - \sum_{j=j1}^{j2} AL(j)\, X(JAL(j))\right)$

adressage indirect + recurrence

TRES DIFFICILE pour le cache !

peu d'opérations / données → "memory bound"

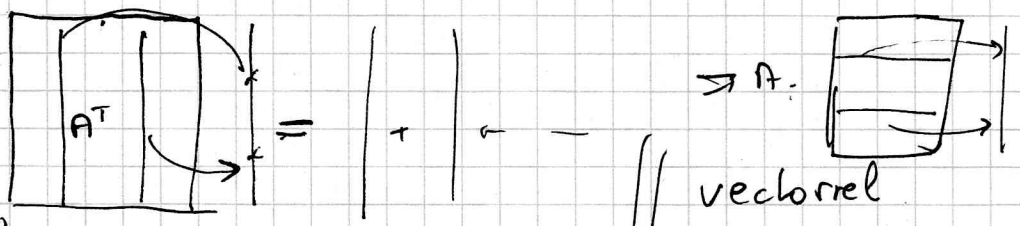
Solution $L^T x = b$ → travailler par colonne

Produit $A^T x$, A CSR

① A "pleine", produit A^T / colonnes $\Rightarrow$ A par lignes

$y = A^T x$,

$$y_j = \sum_{i=1}^{n} A^T_{ji}\, x_i = \sum A_{ij}\, x_i = \sum_{i=1}^{n} x_i\, A_{ij} \qquad j = 1, n$$



Algo

$y = 0$

pour $i = 1 : n$

pour $j = 1 : n$

$y_j = y_j + x_i A_{ij}$

vectoriel

$y = 0$

pour $i = 1, n$

$y = y + x_i\, A(i, :)$

② A CSR, A par lignes

$y = 0$

pour $i = 1 : n$

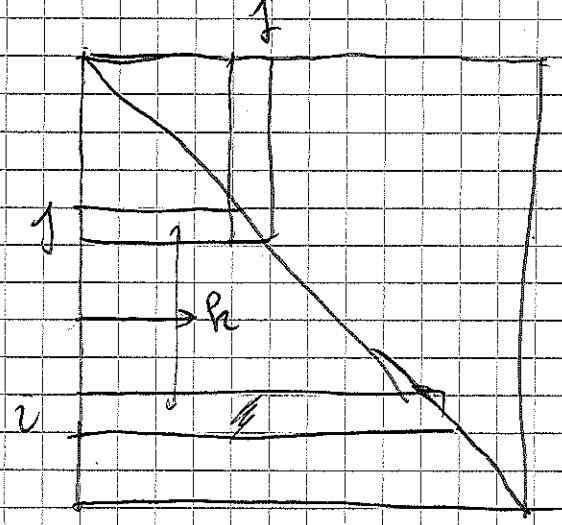
pour $k = IA(i),\ IA(i+1) - 1$

$y(JA(k)) = y(JA(k)) + x(i)\, AA(k)$

Cholesky incomplet pour matrice CSR

Cholesky pour une matrice pleine / ligne (sur place)

```
pour i = 1 : n
    pour j = 1 : i
        pour k = 1 : j-1
            A_ij = A_ij - A_ik A_jk
        si j = i   A_ii = √A_ii
        sinon      A_ij = A_ij / A_jj
```



Matrice creuse symétrique, stockée format CSR, partie inférieure

Tableaux (IA, JA, AA)

Rappel : Les éléments de la ligne I sont stockés dans JA/AA dans l'intervalle [IA(I), IA(I+1)[← ouvert !

et $j = JA(JJ) \Leftrightarrow AA(JJ)$ contient A_{ij}

Pour "aligner" les éléments des lignes i et j en parcourant la ligne j, on doit "inverser" le tableau JA sur la ligne i

Pour cela on introduit IW tq

$IW(J) = JJ$ si $\exists JJ \in [IA(I), IA(I+1)[/ J = JA(JJ)$

$= 0$ sinon

et $AA(IW(K))$ contient alors A_{ik}

Algorithme

```
Pour I = 1 : N
    IW(I) = 0;   J = 1 : N
    J1 = IA(J);   J2 = IA(I+1) - 1;
    Pour JJ = J1 : J2
        IW(JA(JJ)) = JJ
    Pour JJ = J1 : J2
        K1 = IA(JJ);   K2 = IA(JJ+1) - 1
        Pour KK = K1 : K2
            IK = IW(JA(KK))
            Si IK ≠ 0 Alors
                AA(JJ) = AA(JJ) - AA(KK) x AA(IK)
            Si JJ = J2      AA(JJ) = √AA(JJ)
            Sinon           AA(JJ) = AA(JJ) / AA(K2)
```

Méthodes itératives pour les systèmes non symétriques

- GMRES → Arnoldi + moindres carrés
- Autres BiCG → BiCGStab, QMR, (TFQMR)

(I) GMRES

$A \in \mathbb{R}^{n\times n}$, inversible, $b \in \mathbb{R}^n$

$\Rightarrow Ax = b \qquad x \in \mathbb{R}^n$

A pas symétrique $\Rightarrow$ pas de norme d'énergie

[Exemples advection diffusion

$$\begin{cases} -\Delta u + \operatorname{div}(\vec{q}\,u) = f & \text{dans } \Omega \\ u = 0 & \text{sur } \partial\Omega \text{ (ou autres)} \end{cases}$$

$\Rightarrow$ Forme bilinéaire $a(u,v) = \int_\Omega \nabla u \nabla v + \int \vec{q}\,u.\nabla v = \int_\Omega fv$ $\forall v \in H^1_0(\Omega)$

a pas symétrique

Autres Navier Stokes, pbs non linéaires via Newton. matrice Jacobienne pas symétrique

Definition de GMRES

Rappel espace Krylov $K_k(r_0) = \text{vect}(r_0, Ar_0, \dots, A^{k-1}r_0)$

Itération k x_k tq. $\begin{cases} x_k \in x_0 + K_k(r_0) \\ x_k \text{ minimise } \|b - Ax\|_2 \text{ sur } x_0 + K_k \end{cases}$

[Norme euclidienne]

Comme dans le cas symétrique :

$$x \in x_0 + K_k(r_0) \Rightarrow x = x_0 + \sum_{j=0}^{k-1} \gamma_j A^j r_0$$

$$\Rightarrow b - Ax = b - Ax_0 - \sum_{j=0}^{k-1} \gamma_j A^{j+1} r_0 = r_0 - \sum_{j=1}^{k} \gamma_{j-1} A^j r_0$$

C'est-à-dire si $x \in x_0 + K_k(r_0)$, alors le résidu :

$$r = b - Ax = p(A)\, r_0, \quad \text{pour } p \in P_k,\ p(0) = 1$$

(P_k : polynômes de degré $\leq k$)

On en déduit par majoration, pour le résidu r_k de GMRES :

$$\|r_k\|_2 \leq \min_{\substack{p \in P_k \\ p(0)=1}} \|p(A)\, r_0\|_2 .$$

En particulier, pour tout polynôme $p \in P_k$, $p(0) = 1$:

$$\|r_k\|_2 \leq \|p(A)\, r_0\| \leq \|p(A)\|_2 \, \|r_0\|_2$$

Une conséquence est que GMRES termine toujours en au plus n itérations :

On prend pour p le polynôme caractéristique de A

$$p(A) = \det(A - \lambda I) / \det(A) \quad [\det(A) \neq 0 !]$$

⚠ → résultat théorique uniquement

D'après le Th. de Cayley Hamilton, $p(A) = 0 \Rightarrow \|r_n\| = 0$

et x_n est la solution ▨

Mise en œuvre pratique

Supposons connue une base de $K_k(r_0)$, et notons V_k la matrice formée des vecteurs colonnes de cette base. $V_k = [v_1^k, \ldots, v_k^k]$.

La solution à l'étape k s'écrit donc $x_k = x_0 + \sum_{j=1}^{k} y_j v_j^k$

($y \in \mathbb{R}^k$) $\qquad = x_0 + V_k y$

Et donc le problème à résoudre à chaque itération est

$$\min_{y \in \mathbb{R}^k} \|r_0 - A V_k y\|^2 \quad \text{(élevé au carré !)}$$

Cette formulation est peu maniable pour au moins 2 raisons

- L'espace de Krylov $K_k(r_0)$ est un bon espace, mais la base "naturelle" $(r_0, Ar_0, \dots, A^{k-1}r_0)$ donne une matrice V_k très mal conditionnée (explication : méthode de la puissance !)
- Si la matrice V_k n'est pas orthogonale, la norme ne se simplifie pas.

On cherche donc une base orthonormée de $K_k(r_0)$. Cela peut être réalisé par la méthode d'Arnoldi qui consiste à appliquer la méthode de Gram-Schmidt à la base naturelle. Une première version (NE PAS UTILISER, elle sera améliorée plus loin).

Arnoldi : $r_0 = b - Ax_0$, $v_1 = \frac{r_0}{\|r_0\|_2}$

Pour $j = 1, \dots, k$

$h_{jk} = (Av_k, v_j)$

$w_{kn} = Av_k$

pour $j = 1, \dots, k$

$w_{kn} = v_{kn} - h_{jk} v_j$

$h_{kn,k} = \|w_{kn}\|$, si $h_{kn,k} = 0$ STOP

$v_{kn} = \frac{\|w_{kn}\|}{h_{kn,k}}$

Avant de poursuivre, notons que cet algorithme se termine, on a trouvé la solution du problème.

Notons que, tant que $h_{k,k} = 0$, les vecteurs produits par l'algorithme sont orthogonaux (par construction)

Lemme Dans l'algo. précédent, si $h_{k+1,k} = 0$
alors $x^* = A^{-1}b \in K_k(r_0)$

Preuve Par hypothèse, $A v_k = \sum_{j=1}^{k} h_{jk} v_j \in K_k(r_0)$, et
donc $A K_k(r_0) \subset K_k(r_0)$.
Les colonnes de V_k forment une base (orthonormée) de $K_k(r_0)$ donc
$A V_k = V_k H$ où H est une matrice $k \times k$,
H inversible (puisque A l'est)
On a donc $r_k = b - A x_k = r_0 - A(\underbrace{x_k - x_0}_{\in K_k(r_0)})$
$= r_0 - V_k H y$ pour $y \in \mathbb{R}^k$
$\frac{r_0}{\|r_0\|}$ est le premier élément de $K_k(r_0)$, donc $r_0 = \|r_0\|_2 V_k e_1$.
Donc $\|r_k\|_2 = \| \|r_0\|_2 V_k e_1 - V_k H y \|_2$.
Comme V_k est orthogonale, on a $\|r_k\|_2 = \| \|r_0\| e_1 - H y \|_2$
En choisissant y tq $Hy = \|r_0\|_2 e_1$ (H est inversible)
on a donc $\|r_k\|_2 = 0$, et x_k est solution du problème
dans $K_k(r_0)$ ∎

On peut donc supposer que l'algorithme d'Arnoldi ne s'arrête pas.
On définit la matrice $H_k \in \mathbb{R}^{k+1 \times k}$ (rectangulaire) par
$(H_k)_{ij} = (A v_j \; v_i)$
L'algorithme d'Arnoldi peut être résumé matriciellement:

$$A V_k = V_{k+1} H_k$$

(rappel H_k est $k+1 \times k$)

Cette relation, et l'orthogonalité de V_k permettent de simplifier le problème de moindres carrés

On a en effet, avec les notations de la preuve du lemme

$$r_0 = \|r_0\|_2 V_k e_1 = \beta V_k e_1, \quad \beta = \|r_0\|_2, \ e_1 \in \mathbb{R}^{k+1}$$

$$b - A x_k = b - A x_0 - A(x_k - x_0) = r_0 - A(x_k - x_0)$$

$$= \beta V_k e_1 - A V_k y \quad \text{pour } \underline{y \in \mathbb{R}^k, \text{ inconnu}}$$

$$= \beta V_k e_1 - V_{k+1} H_k y$$

et donc

$$\|b - A x_k\|_2 = \|\beta V_{k+1} e_1 - H_k y\|_2 = \|\beta e_1 - H_k y\|_2$$

parce que V_{k+1} est orthogonale.

Le problème se ramène donc, à chaque itération, à

$(MC)_k$ $\quad \min_{y \in \mathbb{R}^k} \|\beta e_1 - H_k y\|_2$

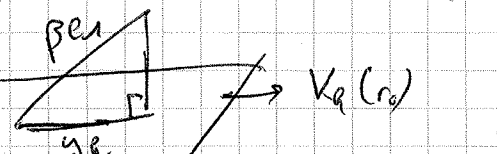
et rappelons que H_k est $(k+1) \times k$, donc un système rectangulaire avec plus d'équations que d'inconnues.

Une fois ce problème résolu (voir plus bas), l'itéré x_k se calcule par $\underline{x_k = x_0 + V_k y_k}$

On voit une première différence importante avec le cas symétrique et défini positif : l'itéré x_k fait intervenir tous les vecteurs v_j, $j = 1, \dots, k$, et ces vecteurs sont aussi nécessaires pour Arnoldi. Le coût de l'itération k augmente avec k.

Résolution du problème de moindres carrés $(MC)_k$

Géométrie



Projection $\perp$ de βe_1 sur $V_k(r_0)$

Résolution par factorisation QR.

Th général $\exists Q \in \mathbb{R}^{(n+k)\times(n+k)}$, Q orthogonale ($Q^TQ = QQ^T = I$) et $R \in \mathbb{R}^{n\times k}$, R_k triangulaire supérieure tq

$$H_k = Q_k \begin{pmatrix} R_k \\ 0 \end{pmatrix} \leftarrow \text{1 ligne}$$

Démonstration constructive (Gram-Schmidt, ou Householder, ou Givens, voir plus loin).

Avec cette factorisation:

$$\|\beta e_1 - H_k y\|_2 = \|\beta e_1 - Q_k \begin{pmatrix} R_k \\ 0 \end{pmatrix} y\|_2$$

$$= \|\beta Q_k^T e_1 - \begin{pmatrix} R_k \\ 0 \end{pmatrix} y\|_2 \quad Q_k \text{ orthogonale}$$

En notant $w_k \in \mathbb{R}^{k+1}$ la première ligne de Q_k, on a, par Pythagore

$$\|\beta e_1 - H_k y\|_2^2 = \|\beta \tilde{w}_k - R_k y\|_2^2 + \beta^2 |(w_k)_{k+1}|^2$$

où $\tilde{w}_k$ comprend les k premiers éléments de w_k.

On en déduit :

1) Le vecteur y est solution de $R_k y = \beta \tilde{w}_k$ (R_k inversible !)

2) Le "résidu" $\|\beta e_1 - H_k y\|_2 = \beta |(w_k)_{k+1}|$ et ceci est égal à $\|b - A x_k\|_2 = \|r_k\|$!!

⚠ Coût de la résolution $O(k^3)$, pour la factorisation si k n'est pas trop grand, et n grand, OK

Conséquence de l'arithmétique en précision limitée

La méthode exposée suppose que les calculs sont réalisés exactement, Si l'on programme Arnoldi sur un ordinateur, en arithmétique flottante (sur 64 bits, 15 chiffres décimaux) on constate que les vecteurs produits ne sont pas orthogonaux.

La modification suivante est donc fondamentale : on remplace l'algorithme de Gram-Schmidt par une version modifiée :

Arnoldi MGS

$r_0 = b - Ax_0$, $v_1 = \frac{r_0}{\|r_0\|_2}$ $k = 0$

$h = h_{kn}$

$w_{kn} = Av_k$

Pour $j = 1, \dots k$

$h_{jk} = (w_{kn}, v_j)$

$w_{kn} = w_{kn} - h_{jk} v_j$

$h_{kn,k} = \|w_{kn}\|_2$

$v_{kn} = \frac{w_{kn}}{h_{kn,k}}$

La seule modification est de fusionner les 2 boucles sur j.

~~Autrement dit elle~~ les 2 algorithmes sont mathématiquement équivalents, mais ont des comportements très différents en arithmétique flottante.

Le premier algorithme ne doit pas être utilisé

En incorporant un critère d'arrêt sur la norme du résidu, on obtient une première version "raisonnable" de GMRES

GMRES avec MGS

$r_0 = b - Ax_0$, $v_1 = \frac{r_0}{\|r_0\|}$, $\rho = \|r_0\|$, $\beta = \rho$, $k = 0$

Tant que $\rho > \varepsilon \|b\|_2$ et $k < k_{max}$ faire

$k = k+1$

$v_{k+1} = A v_k$

pour $j = 1 \to k$

$h_{jk} = (v_{k+1}, v_j)$

$v_{k+1} = v_{k+1} - h_{jk} v_j$

$h_{k+1,k} = \|v_{k+1}\|_2$

$v_{k+1} = v_{k+1} / h_{k+1,k}$

$e_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^{k+1}$

minimiser $\|\beta e_1 - H_k y_k\|_2^2$, pour obtenir $y_k \in \mathbb{R}^k$

$\rho = \|\beta e_1 - H_k y_k\|$

$x_k = x_0 + V_k y^k$

Coût d'une itération : 1 produit matrice vecteur Av_k

k produits scalaires (de taille n)

k "saxpy" ($v_{k+1} - h v_j$), taille n

1 pb (MC) taille $k \to O(k^3)$

Cette dernière étape peut être améliorée.

Multigrille

- Intro blabla, idée
- Problème modèle, analyse spectrale

Pb Dirichlet 1D / 2D

(1D) $\begin{cases} -u'' = f & 0<x<1 \\ u(0) = u(1) = 0 \end{cases}$

Valeurs propres de $-u'' = \lambda u$

$u_k(x) = \sqrt{2}\sin(k\pi x)$ $\quad k = 1, \dots, \infty$, $\quad \lambda_k = k^2\pi^2$

Discrétisation DF $\quad x_i = ih$, $i = 0, \dots, n+1$ $\quad x_0 = 0$, $x_{n+1} = 1$

$h = 1/(n+1)$

$Ax = b$ $\qquad A = \begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}$, $\quad b = h^2 \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$

VP/EV de A : $\quad \theta_k = \frac{k\pi}{n+1}$

$\lambda_k = 4\sin^2\frac{\theta_k}{2}$, $\quad w_k = \begin{pmatrix} \sin\theta_k \\ \sin 2\theta_k \\ \vdots \\ \sin n\theta_k \end{pmatrix} = \begin{pmatrix} \sin(k\pi x_1) \\ \vdots \\ \sin(k\pi x_n) \end{pmatrix}$

cf Fig 13.1 Suad

En 2D $\quad -\Delta u = f$ $\quad]0,1[\times]0,1[$ $\quad h = \frac{1}{n+1}$, $(n+1)\times(n+1)$

Matrice $\quad A = \begin{pmatrix} B & -I & & \\ -I & \ddots & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & B \end{pmatrix}$ $\qquad B = \begin{pmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{pmatrix}$

Valeurs propres $\quad \lambda_{kl} = 4\left(\sin^2\frac{k\pi}{2(n+1)} + \sin^2\frac{l\pi}{2(n+1)}\right)$

vecteurs propres $\quad z_{kl}(x_i, y_j) = \sin(k\pi x_i)\sin(l\pi y_j)$

Analyse spectrale des méthodes itératives

Richardson $\quad x_{k+1} = x_k + \omega(b - Ax_k)$ $\quad$ (gradient à pas fixe)

$= \underbrace{(I - \omega A)}_{M_\omega} x_k + \omega b$

Convergence $0 < \omega < 2/\rho(A) = 2/\lambda_n$

En pratique $\gamma \geq \rho(A)$, $\omega = \frac{1}{\gamma} \leq \frac{1}{\rho(A)} \leq \frac{2}{\rho(A)}$ OK $\omega = \frac{1}{4}$

VP de la matrice d'iteration sont $1 - \omega\lambda_k = 1 - 4\omega \sin^2\frac{\theta_k}{2}$

Erreur décroit comme $e_k = M_\omega^k e_0$ $e_0 = x_0 - x^*$

Dvt sur base des EW: $e_0 = \sum_{j=1}^{n} \xi_j w_j \rightarrow e_k = \sum_{j=1}^{n} \left(1 - \frac{\omega\lambda_j}{\gamma}\right)^k \xi_j w_j$

Chaque composante réduite par $\left(1 - \frac{\omega\lambda_j}{\gamma}\right)$

+ petit facteur $= |1 - \omega\lambda_1|$
+ gd ——— $|1 - \omega\lambda_n|$
$\rightarrow$ $1 - \omega 4 \sin^2\frac{\pi}{2(n+1)} \simeq 1 - \omega\pi^2 h^2$ $\left(\omega = \frac{1}{4}\right)$

$h \rightarrow \frac{h}{2}$ Nbr itérations $\left|\frac{e_k}{e_0}\right| \simeq 10^{-p} = (1 - \omega\pi^2 h^2)^k$

$= 1 - k\omega\pi^2 h^2$

$k \simeq \frac{p \ln 10}{\omega\pi^2} p (n+1)^2$ / $n \rightarrow 2n \Rightarrow$ k multiplié par 4 !!

MAIS l'erreur décroit vite pour certains composantes cf Fig 13.2

Haute fréquence: $k > \frac{n}{2}$

réduction $\eta_k = 1 - \omega 4 \sin^2\frac{k\pi}{2(n+1)}$, pour $\omega = \frac{1}{4}$

$= 1 - \sin^2\frac{k\pi}{2(n+1)} = \cos^2\frac{k\pi}{2(n+1)} \leq \frac{1}{2}$ cf Fig 13.2 dr.

Réduction pour WP oscillants, indépt de h $(= 1/2)$

GRILLE GROSSIERE Taille $2h$, $H = 2h$

n impair $x_i^{2h} = i(2h)$ / $x_i^{2h} = x_{2i}^h$

$j \leq \frac{n}{2}$ $w_j^h(x_{2i}^h) = \sin(j\pi x_{2i}^{2h}) = \sin(j\pi x_i^{2h}) = w_j^{2h}(x_i^{2h})$

mode "lisse" sur grille fine, "infecté" sur grille grossière donne le même mode sur grille grossière. Pour $k \leq \frac{n}{2}$

cf. Fig 13.3

Pour n impair, le mode $w^h_{\frac{n+1}{2}}$ devient le mode le + oscillant

cf

Generalisation : Jacobi relaxé

Jacobi $x_{k+1} = x_k + D^{-1}(E+E^T)\, x_k + D^{-1} b$

Relaxé $x_{k+1} = \omega\left(D^{-1}(E+E^T)\, x_k + D^{-1} b\right) + (1-\omega) x_k$

$= J_\omega x_k + \beta_\omega$

$J_\omega = (1-\omega) I + \omega\, D^{-1}(E+E^T)$; $\beta_\omega = \omega D^{-1} b$

Avec $E + E^T = D - A$, $\underline{J_\omega = I - \omega D^{-1} A}$

Si A SPD alors si $0 < \omega < \frac{2}{\rho(D^{-1}A)}$

Cas modèle $\begin{cases} 1D : & D = 2I \Rightarrow J_\omega = I - \frac{\omega}{2} A \text{ comme Richardson} \\ 2D : & D = 4I \end{cases}$

Cas 1D $s = \sin\frac{\pi}{2(n+1)}$; $\sin^2\frac{k\pi}{2(n+1)} \in [1-s^2,\ s^2]$

$\mu_k(\omega) = 1 - 2\omega \sin^2\frac{k\pi}{2(n+1)}$

$\mu_{k,\ell}(\omega) = 1 - \omega\left(\sin^2\frac{k\pi}{2(n+1)} + \sin^2\frac{\ell\pi}{2(n+1)}\right)$

$1 - 2\omega + 2\omega s^2 \le \mu_k(\omega) \le 1 - 2\omega s^2$

Choix de ω ? Rayon spectral $\rho(J_\omega) = \max\left(|1-2\omega+2\omega s^2|,\ |1-2\omega s^2|\right)$

Si $\omega = \frac{2}{3}$ vp entre $-\frac{1}{3}$ et $\frac{1}{3}$ le mieux possible pour $k > \frac{n}{2}$, $\left(\sin^2\theta_k \ge \frac{1}{2}\right)$

Idem en 2D, meilleur $\omega = 4/5$

Gauss-Seidel Plus compliqué (p 516-517 Cf) Ex 3 ?

Restriction - prolongation

Prolongation : $I_H^h : \Omega_H \longrightarrow \Omega_h$

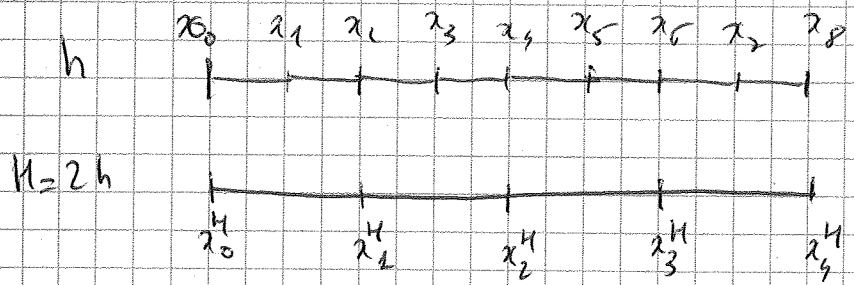
Restriction $I_h^H : \Omega_h \longrightarrow \Omega_H$

Operateurs intergrilles. Restriction / prolongation

Grille $\Omega_h = x_0 = 0 < x_1 = h < \cdots < x_{n+1} = 1$, $x_i = ih$, $i = 0 \cdots n+1$

↳ grille fine — n impair

Grille grossière Ω_H ; $H = 2h$ — 1 point / 2.

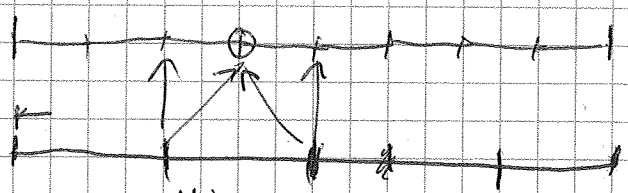


$x_i^H = iH = i(2h) = 2ih = x_{2i}^h$

En général $\Omega_H = x_i^H$, $i = 0, \longrightarrow \frac{n-1}{2}$

Operateur de prolongation : $p : \Omega_H \longrightarrow \Omega_h$

Choix le + simple = interpolation linéaire



vecteur sur Ω_H : $(v_j^H)_{j=0, \frac{n+1}{2}}$ $\qquad v_0^H = v_{\frac{n+1}{2}}^H = 0$

$$\begin{cases} (pv^H)^h_{2jH} = \frac{1}{2}\left(v_j^H + v_{j+1}^H\right) \\ (pv^H)_{2j} = v_j^H \end{cases}$$

$j = 0, \cdots \frac{n-1}{2}$

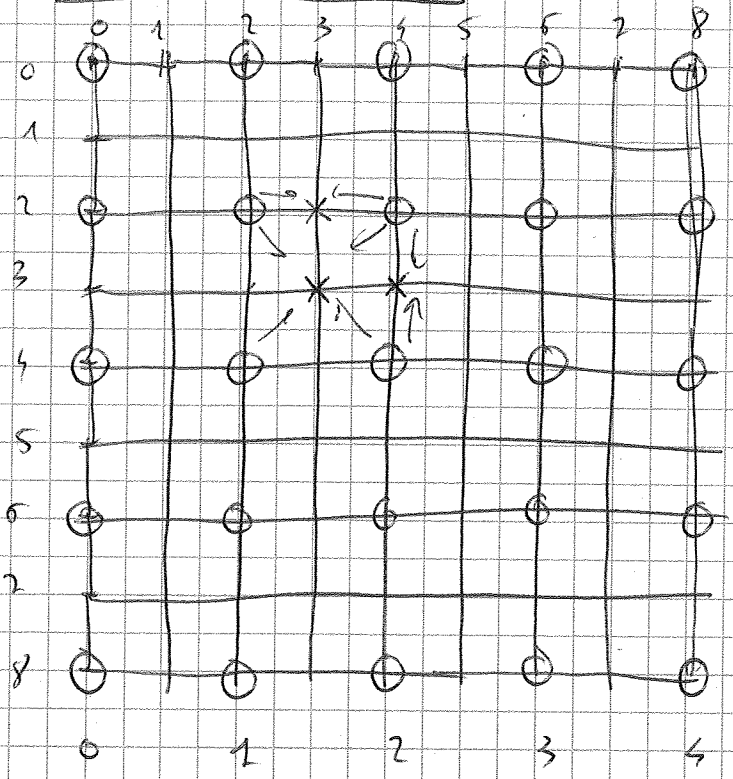
Matrice de p :

$$\begin{array}{c} \\ \frac{1}{2} \\ \\ \end{array} \begin{bmatrix} 1 & & & & \\ 2 & & & & \\ 1 & 1 & & & \\ & 2 & & & \\ & 1 & 1 & & \\ & & 2 & & \\ & & 1 & & \end{bmatrix} \quad \begin{array}{c} \\ \frac{n-1}{2} \\ \\ \end{array}$$

n

Generalisation en 2D

Generalisation en 2D



$$V^h_{2i,2j} = V^H_{ij}$$

$$V^h_{2i+1,2j} = \frac{1}{2}\left(V^H_{ij} + V^H_{i+1,j}\right)$$

$$V^h_{2i,2j+1} = \frac{1}{2}\left(V^H_{ij} + V^H_{i,j+1}\right)$$

$$V^h_{2i+1,2j+1} = \frac{1}{4}\left(V^H_{ij} + V^H_{i+1,j} + V^H_{i,j+1} + V^H_{i+1,j+1}\right)$$

$i = 0, \dots \frac{n_H - 1}{2}$; $j = 0, \dots \frac{n_H}{2}$

(y compris bord)

Restriction $r : \Omega_h \longrightarrow \Omega_H$

Choix naturel : injection simple

$$\Omega_h \longrightarrow \Omega_H$$

$$v^h \longrightarrow v^H = r v^h, \quad v^H_i = v^h_{2i} \quad i = 0, \dots \frac{n-1}{2}$$

mais en général meilleur choix : "Full weighting"

$$v^H_i = \frac{1}{4}\left(v^h_{2i-1} + 2 v^h_{2i} + v^h_{2i+1}\right) \quad i = 0, \dots \frac{n_H}{2}$$

moyenne locale $\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$

Matrice $= \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & \\ & & 1 & 2 & 1 & \\ & & & & 1 & 2 & 1 \end{bmatrix}$

En 2D moyenne sur 9 points. | Stencil $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Remarque 1D : $p = 2r^T$

2D $p = 4r^T$

⚠ En général, on essaie d'avoir $p = r^*$ (opérateur adjoint)

L'adjoint dépend du produit scalaire, et c'est bien ce que l'on observe plus haut.

Pdt scalaire sur Ω_h $(u^h, v^h) = h \sum_{i=1}^{n} u_i v_i = h\, u^T v$

Facteur "h" pour que $(u^h, v^h) \sim \int_0^1 uv\, dx$

Sur la grille grossière $H = 2h$, $(u^H, v^H) = H \sum_{i=1}^{n/2} u_i^H v_i^H = 2h\, u^T v$

Soit $u^h \in \Omega_h$, $v^H \in \Omega_H$

$$(u^h, pv^H) = h\,(u^h)^T (pv^H) = h\,(u^h)^T 2r^T v^H$$
$$= (2h)\,(u^h)^T r^T v^H = H\,(ru^h)^T v^H = (ru^h, v^H)$$

$$\Longrightarrow \boxed{p = r^*}$$

Note : Le cas des éléments finis sera traité plus loin

Définition de l'opérateur "grossier"

Dans le cas du problème modèle, définition naturelle de A_H (DF sur grille de pas H)

En général (mais pas toujours), on définit A_H par

$$A_H = r\, A_h\, p$$

Avantage : si A_h symétrique $\Rightarrow A_H = p^* A_h p$ aussi

Dans le cas du problème modèle, les 2 définitions coïncident (exercice !)

La méthode 2 grilles

Ingrédient : un "lisseur" (méthode itérative classique)

On prend l'exemple de Richardson.

On note S_h la matrice d'itération ($S = I - \omega A$)

Après plusieurs itérations de lisseur, l'erreur (ω "bien choisi") (et le résidu) sont des fonctions lisses (= basse fréquence)

$\Rightarrow$ on les "regarde" sur la grille grossière, et on résout l'équation de correction $A e = d$ sur Ω_H, puis on "remonte" le résultat

v_0^h donné

1) Pré-lissage $v^h = S^{\nu_1} v_0^h$ (+ "2ème membre"), ν_1 itérations
2) Résidu $d^h = b^h - A_h v^h$
3) Restriction $d^H = r\, d^h$
4) Correction grossière $e^H = A_H^{-1} d^H$
5) Prolongation $v^h = v^h + p\, e^H$
6) Post lissage $v_{new}^h = S^{\nu_2} v^h$ (+ 2ème membre), ν_2 itérations

Matrice d'itération : Itérations linéaires, on prend $b^h = 0$

1) $\rightarrow v^h = S^{\nu_1} v_0^h$, 2) $d^H = -(A_h S^{\nu_1}) v_0^h$, 3) $d^H = -(r A_h S^{\nu_1}) v_0^h$

4) $e^H = -(A_H^{-1} r A_h S^{\nu_1}) v_0^h$, 5) $v^h = S^{\nu_1} v_0^h - p A_H^{-1} r A_h S^{\nu_1} v_0^h$

6) Finalement $\boxed{M_h^H = S^{\nu_2} (I - p A_H^{-1} r A_h) S^{\nu_1}}$

On note $M_h^H = S^{\nu_2} T_h^H S^{\nu_1}$, $T_h^H = I - p A_H^{-1} r A_h$

Correction de grille grossière

Méthode multigrille

Suite de grilles Ω_ℓ, $\ell = 0, \dots, L$

Ω_L = niveau fin, pb original $Ax = b \Rightarrow A_L x_L = b_L$

$\Omega_\ell \subset \Omega_{\ell+1}$, on note h_ℓ la "maille" sur Ω_ℓ

opérateur intergrille : $p_\ell : \Omega_{\ell-1} \to \Omega_{\ell+1}$, $r_\ell : \Omega_{\ell+1} \to \Omega_{\ell-1}$

Souvent (pas toujours) $p_\ell = r_\ell^*$

$A_{\ell-1} = r_\ell A_\ell p_\ell$

Idée (du V-cycle) Au lieu de résoudre exactement sur la grille grossière, on applique récursivement le même algorithme.

V-cycle(ℓ, v_0^ℓ, b^ℓ)

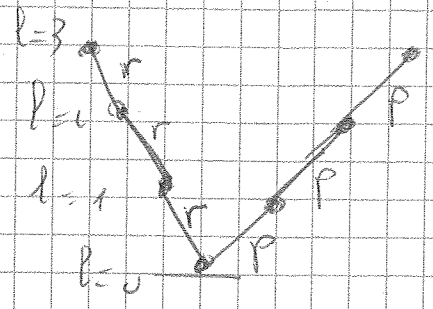
a) Si $\ell = 0$ $v_{new}^\ell = A_0^{-1} b^0$, return

1) $\tilde{v}^0 = v_0^\ell$; for $i = 0, \dots, \nu_1 - 1$: $z^{i+1} = S z^i + g$ (Lissage)
2) $d^\ell = b^\ell - A_\ell z^{\nu_1}$
3) $\hat{d}^{\ell-1} = r_\ell d^\ell$ restriction
4) $e^{\ell-1} =$ V-cycle$(\ell-1, 0, \hat{d}^{\ell-1})$ appel récursif
5) $\hat{v}^\ell = z^{\nu_1} + p_\ell e^{\ell-1}$ correction / prolongation
6) $z^0 = \hat{v}^\ell$, for $i = 0, \dots, \nu_2$: $z^{i+1} = S z^i + g$ post lissage

return z^{ν_2}

Le 4) est un appel récursif. L'algo termine à cause de a)

Illustration V-cycle



Coût d'1 itération de V-cycle :

- n_l nbre éléments sur grille niveau l
  - en 1D $n_l \simeq 2\, n_{l-1}$
  - en 2D $n_l \simeq 4\, n_{l-1}$
- nbre $c_l^{ks} \neq 0$ dans A_l $\quad nnz_l \simeq \leq \alpha\, n_l$
  (ex Laplacien $\alpha \simeq 3$ en 1D, 5 en 2D,)
- Coût d'une itération Passage $\simeq nnz_l \leq \alpha n_l$
- ——— prolongation / restriction $\leq \beta\, n_l$

Au niveau l, le coût est $C_l \simeq \big(\alpha(\gamma_1+\gamma_2) + 2\beta\big) n_l + C_{l-1}$

(γ_1 : passage, β : r/p, C_{l-1} : appel récursif)

En 1D $\quad C_l \simeq \eta\, n_l + C_{l-1}$

avec $n_l \simeq 2^l$

$C_l = \eta\, 2^l + C_{l-1} \longrightarrow C_l \simeq C_0 + \eta \sum_{j=0}^{l} 2^j = C_0 + \eta\, \frac{2^{l+1}-1}{2-1}$

$$C_l \simeq 2^{l+1} \simeq 2\eta\, n_l$$

En 2D idem avec $n_l \simeq 4^l \Rightarrow C_l \simeq \frac{4}{3}\eta\, n_l$

Le coût d'1 itération est proportionnel au nombre d'inconnues

Algo multigrille γ-cycle general

Dans le V-cycle, on remplace l'étape 4/ par

$e^0 = 0$

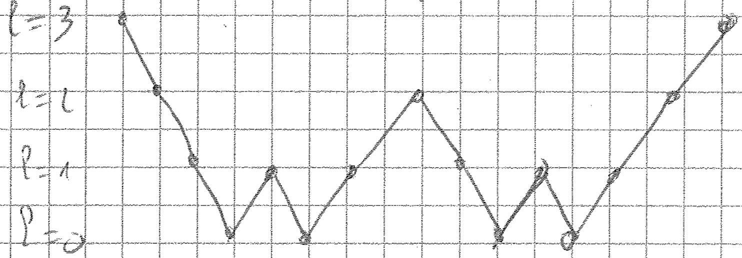
for $i = 0, \dots, \gamma-1$, $e^{i+1} = \gamma\text{-cycle}(\ell-1, e^i, d^{\ell-1})$

et l'étape 5 devient $v^\ell = z^\nu + p_\ell e^\gamma$

En pratique on utilise $\gamma = 1 \rightarrow$ V-cycle (cf + haut)

ou $\gamma = 2$ W-cycle.

Illustration du W cycle ($L = 3$)



Complexité du γ-cycle (coût d'1 itération) :

M hypothèse que pour V-cycle.

$$C_\ell = \eta n_\ell + \gamma C_{\ell-1} \quad (\gamma \text{ appels récursifs})$$

En 2 dimensions, $n_\ell \simeq 4^\ell$. Calcul par une somme en cascade ($\gamma < 4$)

$$C_L = \eta 4^L + \gamma C_{L-1}$$

$$\gamma C_{L-1} = \eta \gamma 4^{L-1} + \gamma^2 C_{L-2}$$

$$\vdots$$

$$\gamma^{L-1} = \eta \gamma^{L-1} 4 + \gamma^L C_0$$

$$\Longrightarrow C_L \leq \eta 4^L \left(1 + \dots + \left(\frac{\gamma}{4}\right)^{L-1}\right) + \gamma^L C_0$$

$$C_L \leq \eta 4^L \, 4 \frac{1}{4-\gamma} + \gamma^L C_0 .$$

$4^L \rightarrow n_L =$ nbre de DOFs sur la grille la + fine

$\gamma = 1$ (algo V), $\simeq \frac{4}{3} \eta n_L + C_0$

$\gamma = 2$ W-cycle $\simeq 2 \eta n_L + 2^L C_0$

coût $\simeq$ cte $\times$ nbre pts sur grille fine.

Pour $\gamma = 4$ $\quad C_k = C_0 + \eta k 4^k \simeq C_0 + \eta n_1 \log n_1$

> Coût $\simeq$ Nbre d'inconnues pour 1 cycle

Matrice d'itération du cycle MG général

Intermède : méthode itérative (preconditionnement) :

x^k, $r^k = b - A x^k$, $\tilde{A}$ inverse approché, $\tilde{A}\delta^k = r^k$ et

$x^{k+1} = x^k + \delta^k = x^k + \tilde{A}^{-1}(b - A x^k)$

On prend ici $x_0 = 0$ $\quad = (I - \tilde{A}^{-1}A)x^k + \tilde{A}^{-1}b = M x^k + \tilde{A}^{-1}b$

Récurrence : $x^k = (I + M + \cdots + M^{k-1})\tilde{A}^{-1}b$

$= (I - M^k)(I - M)^{-1}\tilde{A}^{-1}b$

et on a $M = I - \tilde{A}^{-1}A \Rightarrow (I - M)^{-1}\tilde{A}^{-1} = A^{-1}$

$\rightarrow$ $x^k = (I - M^k)A^{-1}b$

on "remplace" A^{-1} par $(I - M^k)A^{-1}$

Application à la matrice du cycle MG

M_h matrice niveau h, M_H

Dans la matrice à 2 grilles, on remplace A_H^{-1} par $(I - M_H^\gamma)A_H^{-1}$

Ça donne

$$M_h = S_h^{\nu_2}\left[I_h - I_H^h (I_H - M_H^\gamma) A_H^{-1} I_h^H A_h\right] S_h^{\nu_1}$$

$$M_h = M_H^h + S_h^{\nu_2} I_H^h M_H^\gamma A_H^{-1} I_h^H A_h S_h^{\nu_1}$$

= perturbation de 2-grilles

Full-multigrid (= itérations emboîtées)

Obtenir une estimation pour l'itéré initial en partant de la grille grossière

FMG : $h = h_0$, résoudre $A_h u^h = f^h$

Pour $k = 1, \ldots, p$

$u^{h/2} = \hat{I}_h^{h/2} u^h$, $h = h/2$

$u^h = MG^\mu(A_h, u^h, f^h, \nu_1, \nu_2, \gamma)$ $\quad \mu = ?$ $(\mu = 1)$

Full multigrille

Idée : obtenir une estimation pour l'erreur initiale de MG, en partant de la grille la + grossière.

$\hat{P}_\ell$ opérateur de prolongation (ordre + élevé que P)

Algorithme FMG

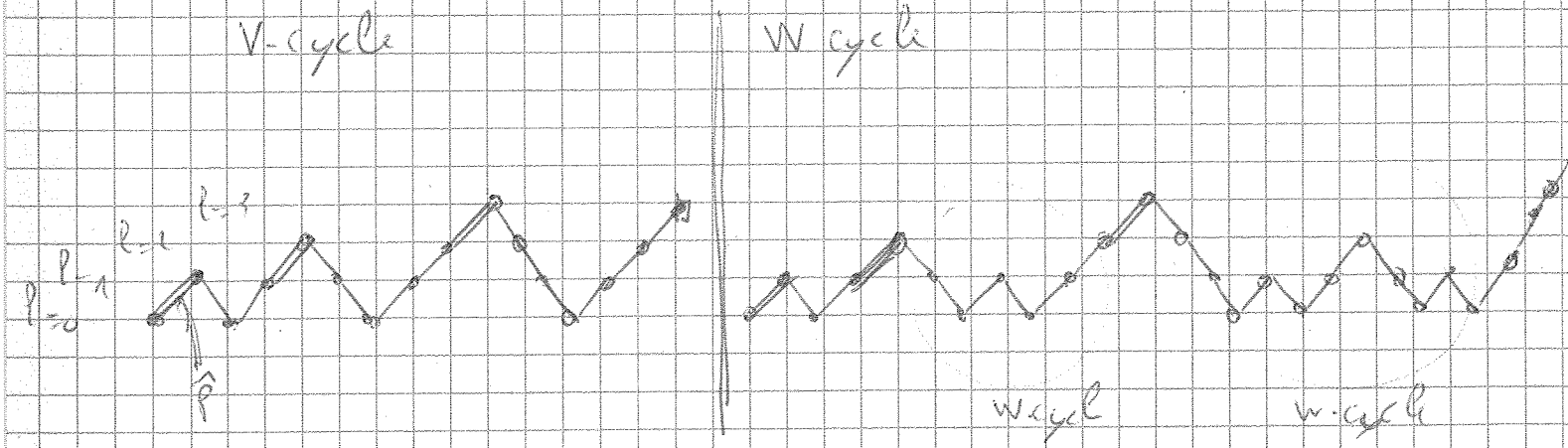
Pour $\ell = 0$ Résoudre $A_0 u^0 = f^0$ (exact)

Pour $\ell = 0, \dots, L-1$

$\hat{u}^\ell = \hat{P}_\ell u^{\ell-1}$

$u^\ell = (MG \gamma)^\mu (\hat{u}^\ell, f^\ell, \nu_1, \nu_2, \bar{\nu})$ [μ itérations]

Illustration



Convergence de la méthode multigrille

Orientation

1) La convergence de la méthode 2-grille, implique la cv du δ-cycle ($\delta \geq 2$) [V-cycle exclu]

2) Pour la méthode 2grille, décomposition de la matrice d'itération, 2 propriété
   - Smoothing property $\Rightarrow$ Lisseur algébrique
   - Approximation ppty $\Rightarrow$ Lié à la relation entre les niveaux Th des EF / admis

3) Résultat vrai pour V-cycle admis

4) Approx optimale pour le Full MG

Rq On parle de convergence indépendante du nombre de niveaux. On montre que la norme (norme 2) de la matrice d'itération est bornée indépendamment de l

---

① Crée 2grilles $\Rightarrow$ crée W-cycle

Cadre variationnel. $\Omega_l = P_l^* \Omega$, $A_{P_l} = P_l^* A P_l$

Hypothèses $\exists$ constante ($\neq$ indpdts l) c_p C_p tq

① $c_p \|v_l\| \leq \|P_l v_{l-1}\| \leq C_p \|v_{l-1}\|$ $\forall v_{l-1} \in V_{l-1}$, $\forall l$

② $\|S^\nu v_l\| \leq C_S \|v_l\|$ $\forall v_l \in V_l$, $\forall \nu$

Pour simplifier, on considère seulement le cas du "pré-smoothing", $\nu_1 = \nu$, $\nu_2 = 0$, et du W-cycle $\delta = 2$

Thm 1 Convergence du W-cycle.

On suppose que la matrice de la méthode 2 grilles vérifie

$$\|M_{l-1}^{\ell}\| \leq \eta(\nu) < 1, \quad \text{avec } \lim_{\nu\to\infty} \eta(\nu) = 0$$

Alors, $\exists \bar{\nu}$ tq $\nu \geq \bar{\nu} \Rightarrow$ $\|M_l\| \leq 2\eta(\nu)$

Ce résultat dit : cv méthode 2 grilles $\Rightarrow$ cvce W-cycle

Démonstration Par récurrence sur l

$l=0$ évident (méthode directe)

Soit $S_l = \|M_l\|$. D'après la récurrence sur M_l :

$$M_l = M_{l-1}^{\ell} + P_l M_{l-1}^{2} A_{l-1}^{-1} P_l^{*} A_l S_l^{\nu}$$

Par hyp $\|M_{l-1}^{\ell}\| \leq \eta(\nu)$,

On remarque que $P_l A_{l-1}^{-1} P^{*} A_l S_l^{\nu} = S_l^{\nu} - M_{l-1}^{\ell}$

et donc $\| \quad \| \leq C_S + 1$,

puis $\|P_l M_{l-1}^{2} A_{l-1}^{-1} P_l^{*} A_l S_l^{\nu}\| \leq C_P S_{l-1}^{2} \|A_{l-1}^{-1} P_l^{*} A_l S_l^{\nu}\|$

$\leq C_l C_P S_{l-1}^{2} \|P_l A_{l-1}^{-1} P_l^{*} A_l S_l^{\nu}\|$

$\leq C_l C_P (C_S + 1) S_{l-1}^{2} = C^{*} S_{l-1}^{2}$

Finalement $S_l \leq \eta(\nu) + C^{*} S_{l-1}^{2}$ et $S_1 \leq \eta$ (hyp)

Si $\bar{\nu}$ tq $\eta(\nu) \leq \frac{1}{4C^{*}}$ pour $\nu \geq \bar{\nu}$ alors $\eta + 4C^{*}\eta^{2} \leq 2\eta$.

Par récurrence, $S_{l-1} \leq 2\eta \Rightarrow S_l \leq \eta + C^{*} 4\eta^{2} \leq 2\eta$ et ok pour $l=1$

CQFD

Analyse de la méthode 2 grille

① "Smoothing property" Mesure de la "smoothness" d'une fonction

$\|A_\ell v_\ell\|_2 = \left(\sum_j \lambda_j |v_{\ell,j}|^2\right)^{1/2}$ (cf modèle problème, Fourier)

Heuristique $\|A_\ell S_\ell^\nu e_\ell\| \ll \|e_\ell\|$

$\Rightarrow$ $\exists \eta(\nu)$, $\lim_{\nu\to\infty} \eta(\nu) = 0$ tq $\|A_\ell S_\ell^\nu\|_2 \le \eta(\nu)\|A_\ell\|_2$ (SP)

Vrai pour Richardson. $S_\ell = I - \omega A_\ell$ $\omega = \frac{1}{\|A_\ell\|_2}$

$A_\ell S_\ell^\nu = A_\ell (I - \omega A_\ell)^\nu = \frac{1}{\omega} B_\ell (I-B)^\nu$ avec $B = \omega A_\ell$

Lemme $\|B(I-B)^\nu\|_2 \le \eta_0(\nu)$, si $\|B\| \le 1$, et B SPD

$\eta_0(\nu) = \frac{\nu^\nu}{(\nu+1)^{\nu+1}} \sim \frac{1}{e\nu} + O\left(\frac{1}{\nu^2}\right)$ $\nu \to \infty$

Rem soit $f(\lambda) = \lambda(1-\lambda)^\nu$ Comme B est SPD

$\|B(I-B)^\nu\|_2 = \max_{\lambda \in \sigma(B)} |f(\lambda)|$

et $|f(\lambda)| \le f\left(\frac{1}{1+\nu}\right) \le \eta_0(\nu)$ (calcul)

Enfin $\frac{\nu^\nu}{(\nu+1)^{\nu+1}} = \left(\frac{\nu}{\nu+1}\right)^{\nu+1} \cdot \frac{1}{\nu} = \frac{1}{\left(1+\frac{1}{\nu}\right)^\nu} \cdot \frac{1}{\nu+1} \sim \frac{1}{e\nu}$ ☐

Donc $\|A_\ell S_\ell^\nu\| \lesssim \|A_\ell\| \eta_0(\nu)$ $\Rightarrow$ (SP) ok pour Richardson ☐

② Approximation property

$\|A_\ell^{-1} - P A_{\ell-1}^{-1} P^*\|_2 \le \frac{C_0}{\|A_\ell\|}$ pour $\ell \ge 1$ (AP)

La preuve de ce résultat d'analyse d'erreur par éléments finis et sera admise ici. Cf Livre de Hackbusch ☐

On définit l'opérateur de restriction par $\underline{r = p^*}$

⚠ la définition du pdt scalaire est celle induite par L^2 sur chaque grille.

Définition des opérateurs à chaque niveau

$$A_{p-1} = r_p A_p p_p = p_p^* A_p p_p$$

de sorte que l'on a bien, pour $v_{p-1}, v_p \in V_{p-1} (\subset V_p)$

$$\begin{aligned} a(v_{p-1}, v_{p-1}) &= a(v_p, v_p) = v_p^T A_p v_p \\ &= (p_p v_{p-1})^T A_p (p_p v_{p-1}) \\ &= v_{p-1}^T \underline{p_p^T A_p p_p} v_{p-1} = v_{p-1}^T A_{p-1} v_{p-1} \end{aligned}$$

et on a la représentation du problème variationnel à chaque niveau.

Le cas des éléments finis

Cadre pb elliptique $V = H_0^1(\Omega)$ (par exemple)

$L : V \to \mathbb{R}$ linéaire continue

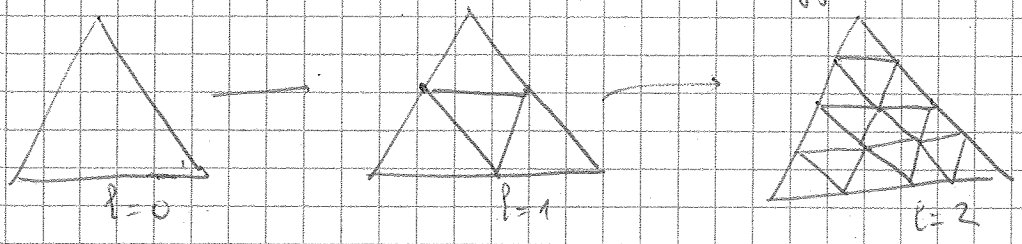
$a : V \times V \to \mathbb{R}$ bilinéaire, continue, coercive

$\exists u \in V \quad a(u,v) = L(v), \ \forall v \in V$ (Lax-Milgram)

Famille de sous espaces $V_\ell \subset V$

Hyp $V_{\ell-1} \subset V_\ell$

En pratique, on part d'une grille grossière, on raffine plusieurs fois. Manière "canonique" de raffiner un triangle



On définit l'op. de prolongation $p : V_{\ell-1} \to V_\ell$ simplement comme injection : une fonction de $V_{\ell-1}$ est aussi dans V_ℓ

⚠ La matrice de p est rectangulaire, en 1D, elle correspond à l'interpolation vue + haut

On note $(\varphi_j^\ell)_{j=1 \to N_\ell}$ les fonctions de base au niveau ℓ.

Par hypothèse, pour $j = 1, \dots, N_{\ell-1}$, on a

$$\varphi_j^{\ell-1} = \sum_{i=1}^{N_\ell} p_{ij}^\ell \varphi_i^\ell$$

et donc, pour $v^{\ell-1} \in V_{\ell-1}$, $v^{\ell-1} = \sum_{j=1}^{N_{\ell-1}} v_j^{\ell-1} \varphi_j^{\ell-1}$

$$= \sum_{j=1}^{N_{\ell-1}} v_j^{\ell-1} \left(\sum_i p_{ij}^\ell \varphi_i^\ell \right)$$

$$= \sum_{i=1}^{N_\ell} \left(\sum_j p_{ij}^\ell v_j^{\ell-1} \right) \varphi_i^\ell$$

soit $v^\ell = P v^{\ell-1}$ matriciellement

Thm² Convergence méthode 2 grilles

Si (SP) et (AP) sont vérifiés la méthode 2 Grilles converg.

$\exists \bar{\nu}$ tq $\nu \geq \bar{\nu} \Rightarrow \|M_{2,1}^{\nu}\| \leq C_0 \eta(\nu)$

Dem On écrit la matrice d'itération de la méthode 2 grilles sous la forme :

$$M_{2,1}^{\nu} = (I - P_\ell A_{\ell-1}^{-1} P_\ell^* A_\ell) S_\ell^{\nu} = (A_\ell^{-1} - P_\ell A_{\ell-1}^{-1} P_\ell^*)(A_\ell S_\ell^{\nu})$$

et $\|M_{2,1}^{\nu}\| \leq C_0 \|A_\ell^{-1}\|^* \times \eta(\nu) \|A_\ell\| \leq C_0 \eta(\nu)$

C@F-D

Convergence du V-cycle (admis)

Hypothèses A_ℓ symétrique, $\nu_1 = \nu_2 = \frac{\nu}{2}$, S_ℓ symétrique + (SP) et (AP)

Thm Le V-cycle converge, avec $\|M_\ell\| \leq \frac{C_A}{C_0 + \nu}$

Ano type d'erreur pour la FMG (Full-multigrid)

App property pour el^{ts} finis   cf Olshanskii

Th pour le V-cycle Π hypothèses

Vcycle avec $\nu_1 = \nu_2$ converge $\forall \nu$!

$\|M_h\| \leq C < 1$

Convergence pour FMG

1) Erreur EF $\|u - u^h\| \leq C h^2$ par exemple

2) Interpolation FMG: $\|u^h - \hat{I}_H^h u^H\| \leq c_1 h^K$ (K)?

3) $\|M_h\| \leq \xi < 1$ cf avant

3) $\|\hat{I}_H^h\| \leq c_2 2^{-K}$

(Th) (cf Saad, Th 13.2) μ assez gd pour que $c_2 \xi^\mu < 1$

Alors $\|u^h - \tilde{u}^h\| \leq C h^K$ $C = \dfrac{c_1 \xi^\mu}{1 - c_2 \xi^\mu}$

(u^h exact, $\tilde{u}^h$ FMG)

Rem : Th + complexité ⇒ optimal = Discrétisation erreur pour 1 coût $\propto$ nbre inconnus

Dem Récurrence.

Niveau + grossier : OK, car $\tilde{u}_h$ sol exacte

Th OK pour H ⇒ OK pour h

$$u^h - \tilde{u}^h = (M_h)^\mu (u^h - u_0^h), \quad u_0^h = \hat{I}_H^h \tilde{u}^H$$

$$\|u^h - u_0^h\| = \|u^h - \hat{I}_H^h u^H + \hat{I}_H^h (u^H - \tilde{u}^H)\|$$

$$\leq \|u^h - \hat{I}_H^h u^H\| + \|\hat{I}_H^h (u^H - \tilde{u}^H)\|$$

$$\leq c_1 h^K + c_2 2^{-K} C H^K = h^K (c_1 + C c_2)$$

$$c_1 + C c_2 = c_1 + \frac{c_1 c_2 \xi^\mu}{1 - c_2 \xi^\mu} = \frac{c_1}{1 - c_2 \xi^\mu}$$

$$\Rightarrow \|u^h - \tilde{u}^h\| \leq \frac{c_1 \xi^\mu}{1 - c_2 \xi^\mu}$$ OK ■

$\mu = 1$ suffit