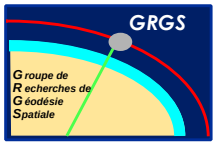


Méthodes de résolution des équations normales, ou d 'observations

G. Balmino , CNES-GRGS

Ecole d' été du GRGS
« Détermination du champ de gravité terrestre par les nouvelles missions spatiales »

Forcalquier, 2-6 Septembre 2002

Première Partie

RESOLUTION EXPLICITE DES EQUATIONS

Seconde Partie

METHODES ITERATIVES ET SEMI-ITERATIVES

Logiciel DYNAMO
(modules D, DGC,...)

(cf. cours de J.M.Lemoine, J.C. Marty)

Première Partie

RESOLUTION EXPLICITE DES EQUATIONS

1. Résolution classique des équations normales :

. méthode de Cholesky

2. Résolution à partir des équations d ’observations :

. décomposition Q-R

(algos. : Householder, Givens)

. combinaison de décompositions

Solution des problèmes de moindres carrés de grande taille

Système : **m** équations , **n** inconnuess , **m** > **n**

$\underline{A}\, x = \underline{b}$; *matrice de poids* : Π

$\Leftrightarrow$ $A\, x = b$; $A = \Pi^{1/2}\, \underline{A}$; $b = \Pi^{1/2}\, \underline{b}$

Solution par moindres carrés

1. Classique : Equations normales: $A^T A\, x = A^T b$ (*s.d.p.*)

puis décomposition de Cholesky de $N = R^T_N \,.\, R_N$

2. Par transformations orthogonales : $A = Q\, R$

Q : *orthogonale* , $= Q_1 \,.Q_2 \,. \ldots Q_r$ $(r \leq \mathrm{n})$ and $R = \begin{bmatrix} R^* \\ 0 \end{bmatrix}$ $(R^* = R_N)$

Chaque Q^T_k : matrice de Householder ou Givens

Puis : $R^{*T}\, y = A^T\, b \rightarrow y$, et $R^*\, x = y \rightarrow x$, et $cov(x) = R_N^{-1}\, R_N^{-T}$

$v = (v_i)$ *est un* **n**-*vector*

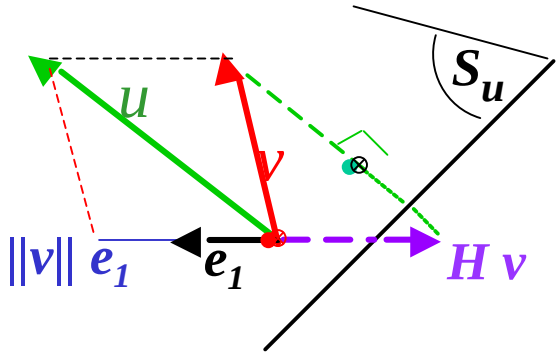
Householder

$$e_1 = [1\ 0\ 0\ \ldots\ 0]^T$$

$$u = v + \varepsilon \| v \| \ e_1 \ ; \ \ (\varepsilon = +1 \ \text{si} \ v_1 \geq 0 \ , \ \varepsilon = -1 \ \text{si} \ v_1 < 0)$$

$$H = I_n - 2\, u\, u^T / (u^T u) \ \rightarrow \ H\, v = - e \| v \| \ e_1$$

= Symétrie / sous-espace S_u **,**dim.= **n**-1, orthogonal à ***u*** : $Hu = -u$; $Hs = s$ $(s \in S_u)$



Givens

$$G_{ij} = \delta_{ij} [1+(c-1) (\delta_{iI} + \delta_{jJ})] + (s-1) (\delta_{iI}\, \delta_{jJ} - \delta_{iJ}\, \delta_{jI})$$

avec $c = v_I / (v_I^2 + v_J^2)^{1/2}$, $s = v_J / (v_I^2 + v_J^2)^{1/2}$

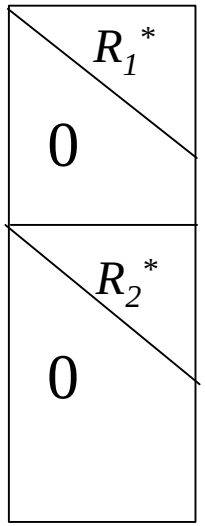
= Rotation dans le plan $\{e_I , e_J\}$: $\quad v_I \rightarrow 1 \ ; \ v_J \rightarrow 0$

Combinaison de matrices R

(= *analogie avec combinaison de systèmes normaux)*

$$A_1 x = b_1 \rightarrow R_1 x = Q_1^T b_1$$
$$A_2 x = b_2 \rightarrow R_2 x = Q_2^T b_2$$

$$\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} x = \begin{bmatrix} Q_1^T b_1 \\ Q_2^T b_2 \end{bmatrix} = b_{12}$$



G : plus adapté

$$Q_{12} R_{12} x = b_{12}$$

etc ...

Avantages -inconvénients des deux méthodes

Eq. normales + Cholesky

Avantages

- facilités de programmation
- existence de logiciels en bibliothèques « out-of-core » (+ Cholesky par blocs)

Inconvénients

- temps de formation eq. norm. **m n²/2** op. (+*)
- perte de précision (à la formation)

Eq. d ’observ. + *QR*

Avantages

- grande précision

Inconvénients

- temps de calcul : **m n²** op.
- peu d ’expérience avec les très grands systèmes
- programmation complexe pour syst. de rang r < n, surtout en « out-of-core »

METHODES ITERATIVES ET SEMI-ITERATIVES

I- METHODES ITERATIVES GENERALES

1. Principes
2. Trois méthodes générales
   - *2.1. Jacobi (J)*
   - *2.2. Gauss-Seidel (GS)*
   - *2.3. Sur-relaxations successives (SOR)*

II- METHODES DE GRADIENT POUR SYSTEMES S.D.P.

1. Principes
2. Descente rapide (SD)
3. Gradients conjugués (CG)
4. Adaptation de CG aux problèmes de moindres carrés

III- APPLICATIONS :

1. Rappels
2. Méthode de Jacobi par blocs (BJ)
3. Gradients conjugués préconditionnés (PCG)
4. Adaptations de BJ et PCG au traitement direct des équations d ’observation
5. Méthodes BJ - *comb* et PCG - *comb* dans un contexte de parallélisation

Quelques remarques en conclusion

I - METHODES ITERATIVES GENERALES

1. PRINCIPES

$n \sim 10^3 , 10^4 , 10^5 , \ldots$

$Ax = b$: $A\ (n \times n)$, $b\ (n \times 1)$, $x\ (n \times 1)$

Attention ! Notations différentes de la 1.ère partie
A = matrice normale

Calcul direct de A^{-1} : temps $\sim o\ (n^3)$ + pb. Mémoire centrale *vs.* I/O

Supposons : $M \approx A$ (à préciser) ; M simple (facile à inverser)
x_0 donné (par ex. $x_0 = M^{-1} b$) ; on pose $x = x_0 + \Delta x$

$$\Rightarrow \quad M x = Mx - Ax + b$$
$$= (M-A)(x_0 + \Delta x) + b = (M-A)\, x_0 + b + \underbrace{(M-A)\, \Delta x}_{\textit{(négligé)}}$$

D 'où : $M x_1 = (M-A)\, x_0 + b$

. . .

$M x^{(k+1)} = (M-A)\, x^{(k)} + b$

ou : $x_{k+1} = M^{-1} [\, (M-A)\, x_k + b]$
$= x_k - M^{-1} r_k$ (avec $r_k = A x_k - b$: résidu)

2 écritures :
$\mathbf{x^{(k)}}$ ***ou*** $\mathbf{x_k}$

Convergence du processus

$$M\,x = (M-A)\,x + b$$
$$M\,x_{k+1} = (M-A)\,x_k + b$$

$$\rightarrow \quad M\,(x - x_{k+1}) = (M - A)\,(x - x_k)$$

ou : $M\ e_{k+1} = (M - A)\ e_k$

donc $e_{k+1} = (I - M^{-1} A)\ e_k$

$= B$: matrice de « décision » (critique)

$$\Rightarrow \quad e_k = B^k\, e_0$$

Si $B = P\,\Lambda\,P^{-1}$, avec $\Lambda = \mathrm{diag}\,(\lambda_i)_i$, $\lambda_i \in \mathbb{C}$

$B^k = P\,\Lambda^k\,P^{-1}$

D 'où : $(\Lambda^k \rightarrow \mathbf{0}) \Rightarrow (B^k \rightarrow \mathbf{0}) \Rightarrow (e_k \rightarrow 0)$

$\Rightarrow \quad x_k \rightarrow x$

Il faut $|\lambda_i| < 1$

Rayon spectral de B : $|\lambda_i|_{max} = R$

L 'estimation de *R* est difficile (bornes seulement, en général)

Calcul de l ’inverse de A

Dans l ’équation de base on fait les substitutions suivantes :

$$x \rightarrow A^{-1}$$
$$b \rightarrow I$$

donc : $Ax = b \rightarrow A\,A^{-1} = I$

D ’où le schéma itératif :

$$A^{-1}_{k+1} = A^{-1}_{k} - M^{-1}\,(A\,A^{-1}_{k} - I)$$

Choix de M

M = diagonale de A : Jacobi

M = triangle inférieur de A : Gauss-Seidel

M = triangle inférieur de A
avec diagonale modifiée : Méthode(s) de relaxation

Paragraphe suivant

2.1. Jacobi

$$M = D = \text{diag}\,(a_{ii})_i$$

$$a_{ii}\ x_i^{k+1} = -\underbrace{\sum_{j=1}^{i-1} a_{ij}\ x_j^k}_{\textit{si } i > 1} - \underbrace{\sum_{j=i+1}^{n} a_{ij}\ x_j^k}_{\textit{si } i > n} + b_i$$

Les valeurs propres m_i de $B_J = I - D^{-1} A$ sont de module < 1.

$\Rightarrow$ $e^k \longrightarrow 0.$

CV rapide si A est à diagonale dominante

Jacobi est bien adapté à la parallélisation

Les calculs sur les lignes sont indépendants d 'une ligne à l 'autre.

2.2. Gauss-Seidel

Idée | Schéma proche de celui de Jacobi,
mais x_j^{k+1} ($j<i$) est immédiatement utilisé pour calculer x_i^{k+1}

i.e.
$$a_{ii}\ x_i^{k+1} = -\sum_{j=1}^{i-1} a_{ij}\ x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}\ x_j^{k} + b_i$$

$$\Longleftrightarrow \quad \sum_{j=1}^{i} a_{ij}\ x_j^{k+1} = -\sum_{j=i+1}^{n} a_{ij}\ x_j^{k} + b_i$$

ou :
$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \end{bmatrix} \begin{pmatrix} x^{k+1} \end{pmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{M\text{-}A} \end{bmatrix} \begin{pmatrix} x^{k} \end{pmatrix} + \begin{pmatrix} b \end{pmatrix}$$

M = partie triangulaire inf. de **A**

triangle sup. strict
(diag.=0)

⇒ Procédure séquentielle : pas de //ation … mais CV 2* plus rapide que Jacobi

2.3. Sur-relaxations successives (SOR)

Successive Over-Relaxation

G.S. $$x_i^{k+1} = \frac{1}{a_{ii}} \left\{ - \sum_{j=1}^{i-1} a_{ij}\ x_j^{k+1} \quad - \sum_{j=i+1}^{n} a_{ij}\ x_j^{k} + b_i \right\}$$

Idée : Corriger « un peu plus » $x_i^{k+1} - x_i^{k}$ par un facteur $w > 1$.

$$\begin{aligned} & a_{i1}\, x_1^{k+1} + a_{i2}\, x_2^{k+1} + \ldots a_{i,i-1}\, x_{i-1}^{k+1} + \frac{a_{ii}}{w}\, x_i^{k+1} \\ & = - a_{i,i+1}\, x_{i+1}^{k} - a_{i,i+2}\, x_{i+2}^{k} - \ldots \quad + b_i \\ & \quad + \frac{a_{ii}}{w}\, x_i^{k+1} - a_{ii}\, x_i^{k+1} \end{aligned}$$

remplacés par x_i^k à l 'étape en cours $(k+1)$

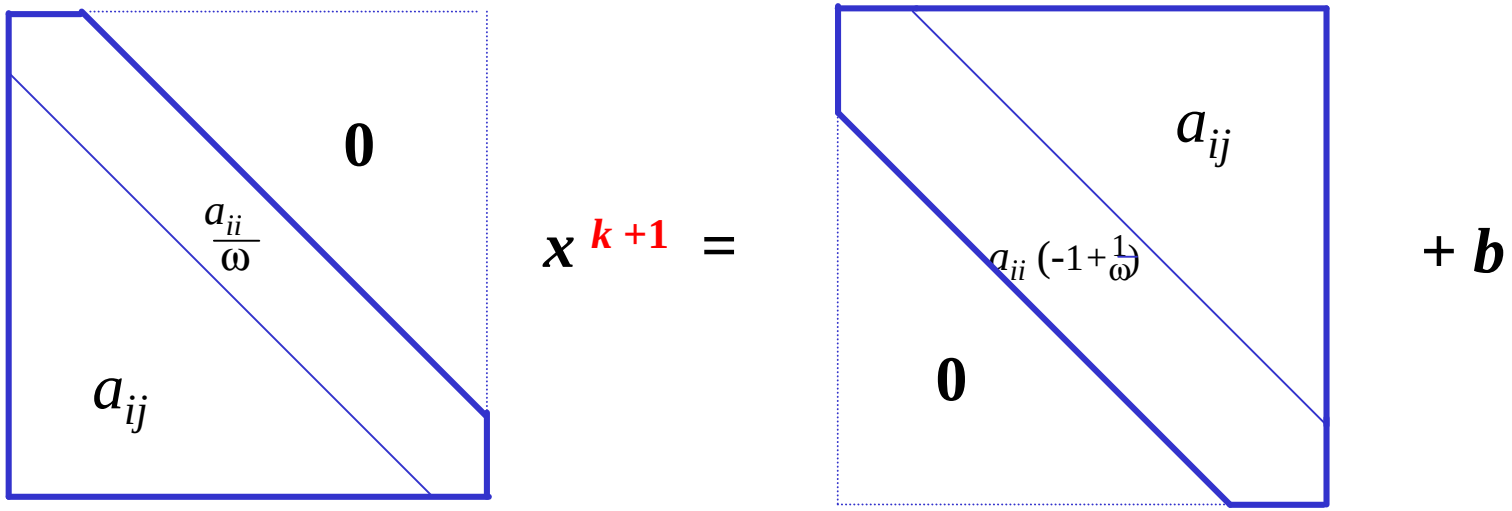
$$\begin{aligned} & a_{i1}\, x_1^{k+1} + a_{i2}\, x_2^{k+1} + \ldots a_{i,i-1}\, x_{i-1}^{k+1} + \frac{a_{ii}}{w}\, x_i^{k+1} \\ & = (-1+\tfrac{1}{w}) a_{ii} x_i^{k} - a_{i,i+1}\, x_{i+1}^{k} - a_{i,i+2}\, x_{i+2}^{k} - \ldots \quad + b_i \end{aligned}$$

Par suite :

GS ⟶ **SOR** ω

M ⟶ M_ω : a_{ij} $(j \le i)$ remplacé par $a_{ij}^{(w)} = a_{ij} / w^{d_{ij}}$

M-A ⟶ $(M\text{-}A)_\omega$: $-a_{ij}$ $(j \ge i)$ remplacé par $-a_{ij}^{(w)} = (-1 + \frac{d_{ij}}{w}) a_{ij}$



Choix de ω

$$B_{SOR} = I - M_w^{-1} A = M_w^{-1} (M_w - A)$$

$\Rightarrow$ *déterminants :* $\quad |B_{SOR}| = |M_w^{-1}| . |M_w - A| = |M_w|^{-1} . |M_w - A|$

avec : $\quad |M_w| = \frac{1}{w^n} \prod_{i=1}^{n} a_{ii}$

$$|M_w - A| = \left(\frac{1-w}{w}\right)^n \prod_{i=1}^{n} a_{ii}$$

Donc : $\quad$ ***abs*** $|B_{SOR}| = |(1-w)^n| = \prod_{i=1}^{n} |\lambda_i| < 1$ *(pour CV)*

avec $\omega > 1$ $\longrightarrow$ $\omega < 2$

(Dans la pratique w=**1.9 *est un bon choix...)***

Remarque

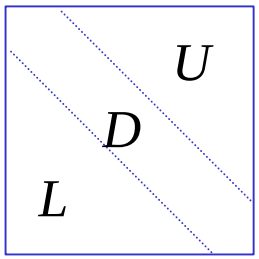
On a : $\quad l_{\max}^{(SOR)} \leq \frac{2}{1+\sqrt{1-m_{\max}^2}} - 1$

avec $\quad m_{\max} = l_{\max}^{(Jacobi)}$

...et $\quad w_{\text{optimal}} = l_{\max}^{(SOR)} + 1 \quad$ (Young, 1950)

RESUME

On écrit : $A = L + D + U$



⇒ Schéma itératif :

$$\boldsymbol{F}\, x^{k+1} = \boldsymbol{G}\, x^{k} + b \;,\; (k = 0, 1, 2, \ldots \boldsymbol{n}_{iter})$$

Méthode	F	G
Jacobi	D	$-L-U$
Gauss-Seidel	$L + D$	$-U$
SOR	$L + D/\omega$	$(-1+1/\omega)D - U$

L ’intérêt est évidemment d ’avoir une solution satisfaisante avec un nombre $\boldsymbol{n}_{iter}$ d ’itérations petit devant n .

II - METHODES DE GRADIENT POUR SYSTEMES s.d.p.

1. PRINCIPES

(cf. GINS, DYNAMO)

$$C\,x = d \qquad (1)$$

s.d.p. : sym., (semi-) définie, positive

i.e. $x^T C\,x \geq 0$

Résoudre (1) est équivalent à chercher le minimum de

$$\Phi(x) = \tfrac{1}{2}\,x^T C x - x^T d \qquad (2)$$

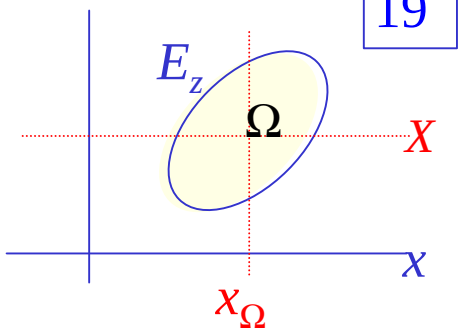
En effet : $A\,x = b$ *, avec* $A \leftarrow \sqrt{\Pi}\,A$
possède la solution x *telle que*
$2\,\Psi(x) = (Ax-b)^T\,(Ax-b)$: *minimum*
Or $\Psi(x) = 1/2\;x^T \underbrace{A^T A}_{=C}\,x - x^T \underbrace{A^T b}_{=d} + 1/2\;b^T b = \Phi(x) + \underbrace{1/2\;b^T b}_{>0}$

(2) minimum $\Rightarrow$ $d\,\Phi(x) = \tfrac{1}{2}\,(dx^T\,Cx + x^T C\,dx) - dx^T\,d$

$$= dx^T\,(Cx - d) = 0$$

Donc : $\{\Phi'(x) = grad(\Phi) = 0\} \quad \Leftrightarrow \quad \{Cx - d = 0\}$

Si C est s.d.p., $\Phi(x) = z$ définit un ellipsoide E_z de $\mathbf{R}^n$



Tous les ellipsoides ont :

- mêmes directions d 'axes (*vecteurs propres de C*)
- même centre Ω ⟶ x_Ω ?

On pose $x = x_\Omega + X$

⇒ *supprimer les termes en X (i.e. linéaires)*

Donc :

$$\Phi(x) = \Phi(x_\Omega + X) = \tfrac{1}{2}\, X^T C X + X^T (C x_\Omega - d) + \tfrac{1}{2}\, x_\Omega{}^T (C x_\Omega - d) - \tfrac{1}{2}\, x_\Omega{}^T d$$

⇒ Ω *défini par :* $\boxed{C x_\Omega - d = 0}$

et $\Phi(x)$ *est minimum au centre* Ω , *i.e. en* $X=0$

... car $\Phi(X) = \tfrac{1}{2}\, \underbrace{X^T C X}_{\geq 0} - \tfrac{1}{2}\, d^T C^{-1} d$

Le problème revient à trouver le centre des ellipsoides E_z par approximations successives

⟹ *Cheminement suivant un trajet* $\boldsymbol{p_0}$, $\boldsymbol{p_1}$, $\boldsymbol{p_2}$, ...

$x_0 \xrightarrow{\boldsymbol{p_0}} x_1 \xrightarrow{\boldsymbol{p_1}} x_2 \xrightarrow{\boldsymbol{p_2}} x_3$ *etc...*

TECHNIQUE

- on part de $x_0 \Rightarrow C\,x_0 - d = r_0$
- on se donne p_0 (direction de « pente ») ……. *on verra le choix de* p_0
- on pose : $x_1 = x_0 + t_0\, p_0$

le facteur d 'échelle t_0 est tel que $\Phi(x_1\,(t_0))$ est minimum

(on aura donc un mimimum dans la direction de p_0*)*

Þ $\Phi(x_1\,(t_0)) = 1/2\; x_1^T\, C\, x_1 - x_1^T\, d$

$= 1/2\; (x_0 + t_0\, p_0)^T\; C\; (x_0 + t_0\, p_0) - (x_0 + t_0\, p_0)^T\; d$

$= \Phi(x_0) + 1/2\; t_0^2\, p_0^T\, C\, p_0 + t_0\, r_0^T\, p_0$

$\Phi(x_1\,(t_0))$ ***min.*** si :

$$\boldsymbol{t_0\; p_0^T\, C\, p_0 + r_0^T\, p_0} = 0 \qquad (3)$$

D 'où : $t_0 = - r_0^T\, p_0 \,/\, (p_0^T\, C\, p_0)$

p_0 *non* $\perp$ *à* r_0 *, sinon* $x_0 \equiv x_1$

On a bien $\Phi(x_1)$ *min. , car* $d^2\Phi / dt_0^2 = p_0^T\, C\, p_0 > 0.$

De plus : $\Phi(x_1) < \Phi(x_0)$

en effet $\Phi(x_1) - \Phi(x_0) = 1/2\; t_0^2\, p_0^T\, C\, p_0 + t_0\, r_0^T\, p_0$

$= 1/2\; (r_0^T\, p_0)^2 \,/\, (p_0^T\, C\, p_0) - (r_0^T\, p_0)^2 \,/\, (p_0^T\, C\, p_0)$

$= - 1/2\; (r_0^T\, p_0)^2 \,/\, (p_0^T\, C\, p_0) < 0.$ *(QED)*

Propriétés :

- $grad\ \Phi(x_1) = r_1$

$\ldots\ldots$ *En effet* $\Phi\,'(x_1) = C\,x_1 - d = r_1$

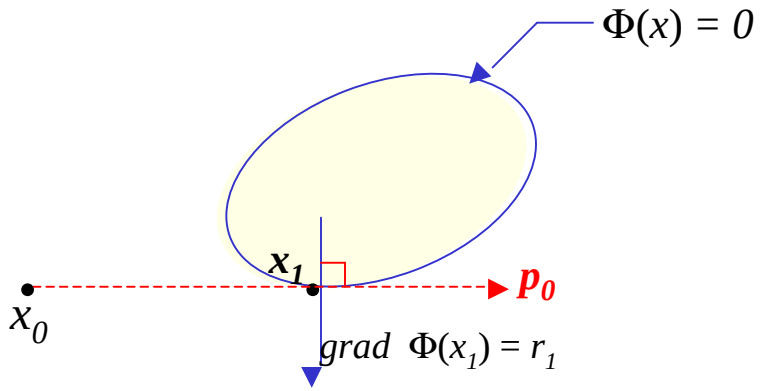
- $r_1 \perp p_0$

$\ldots\ldots$ $r_1 = C\,x_1 - d = C\,(x_0 + t_0\,p_0) - d = r_0 + t_0\ C\,p_0$

$\Rightarrow$ $p_0^T\ r_1 = p_0^T\ r_0 + t_0\ p_0^T\,C\,p_0$

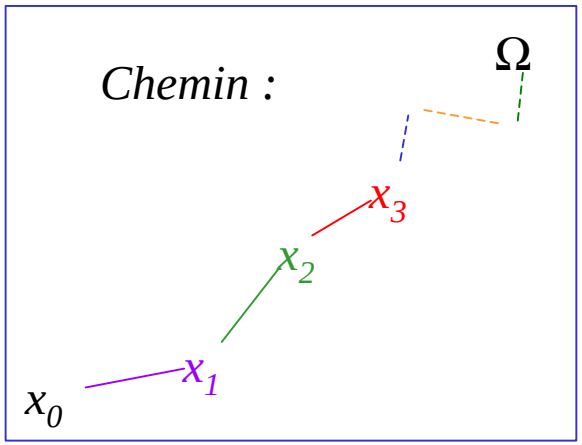
$= 0.$ $\ldots\ldots$ *d 'après (3)*

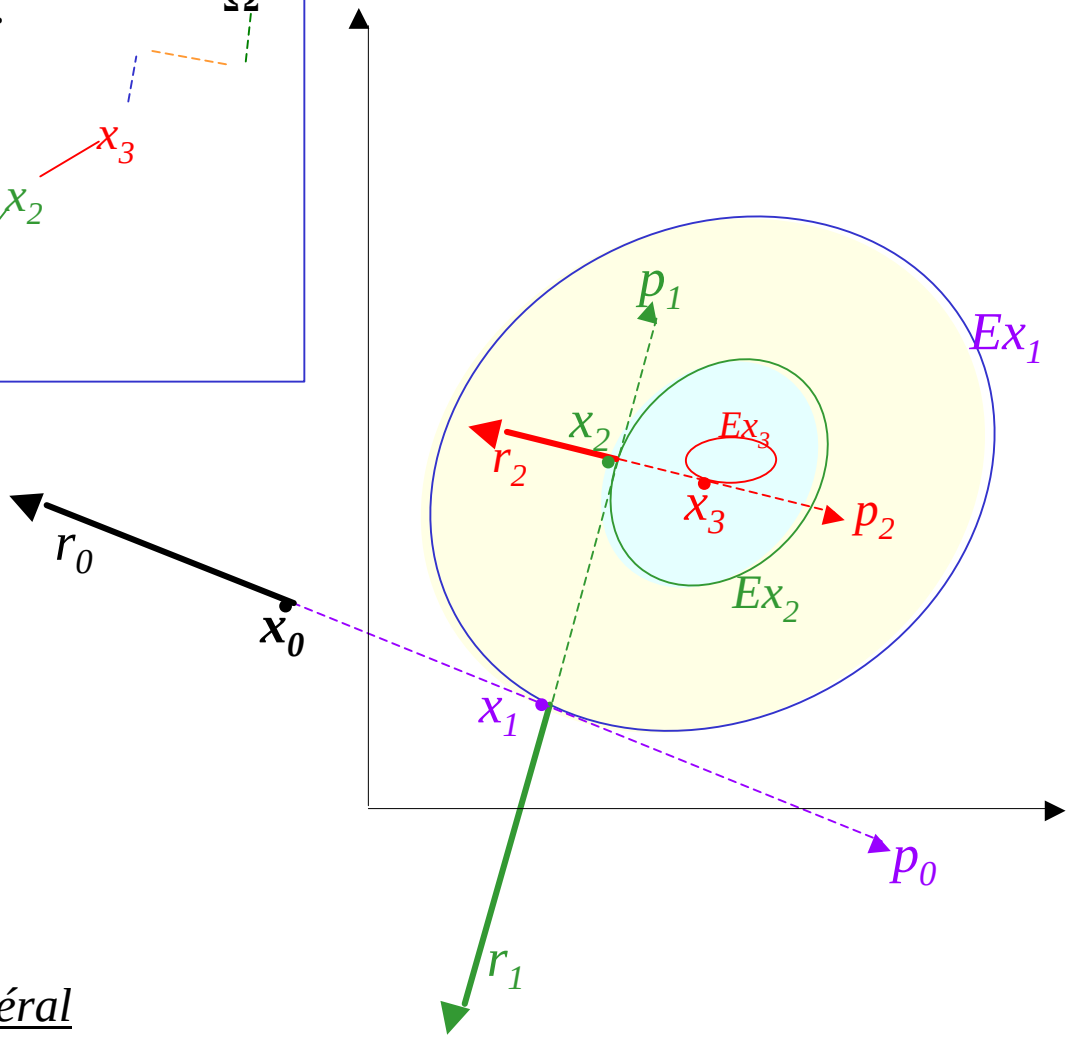
Ces deux propriétés impliquent que p_0 est tangent à l 'ellipsoide passant par x_1



2. DESCENTE RAPIDE (SD = *steepest descent*) :

Cas particulier de la méthode générale





p_0 *tangent à* Ex_1
p_1 *tangent à* Ex_2
p_2 *tangent à* Ex_3
…
$\Leftrightarrow p_k \perp r_{k+1}$

Cas général
p_k *non* $\perp$ *à* r_k

Cas particulier (SD)
$p_k = - r_k$
(cas de la figure)

Cas général

. x_0
. $r_0 = C\,x_0 - d$
. p_0 *non* $\perp$ *à* r_0
. $t_k = f(p_k)$: $x_{k+1} = x_k + t_k\ p_k$

choix

$$t_k = - r_k^T\ p_k / (p_k^T\ C\ p_k)$$

. $r_{k+1} = r_k + t_k\ C\ p_k$

(et on vérifie que $r_{k+1}^T\ p_k = 0$*.)*

Cas particulier (Steepest Descent)

. x_0
. $r_0 = C\,x_0 - d$
. $p_0 = - r_0$, … $p_k = - r_k$ (*)
. $x_{k+1} = x_k + t_k\ p_k$

avec $t_k = r_k^T\ r_k / (r_k^T\ C\ r_k)$

. $r_{k+1} = r_k + t_k\ C\ p_k$

(et on vérifie que $r_{k+1}^T\ p_k = 0$*.)*

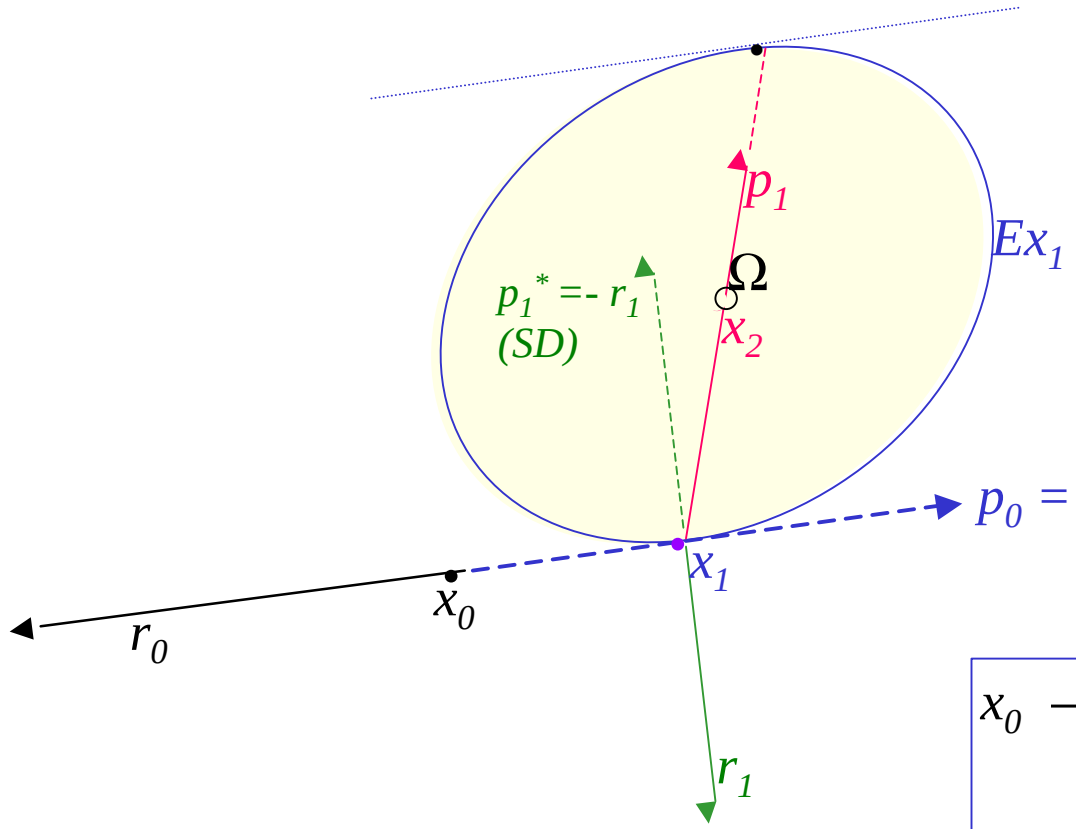
(*) *Justification :* $\Phi(x)$ *est* *min.* *en* Ω *et grad* $\Phi(x_k) = r_k$.
Or le vecteur grad Φ *est dirigé vers les valeurs de* Φ ***croissantes.***
Pour aller vers le minimum il faut donc aller dans le sens opposé
$\Rightarrow$ $p_k = - r_k$ *, et* t_k *est évidemment* ≥ 0 *d 'après son expression.*
On reste donc bien dans le bon sens.

3. GRADIENTS CONJUGUES (CG = *Conjugate Gradients*)

Principe : $p_0 = - r_0$

$[p_{k-1}, p_k]$: *conjugués* → Exemples

(a)



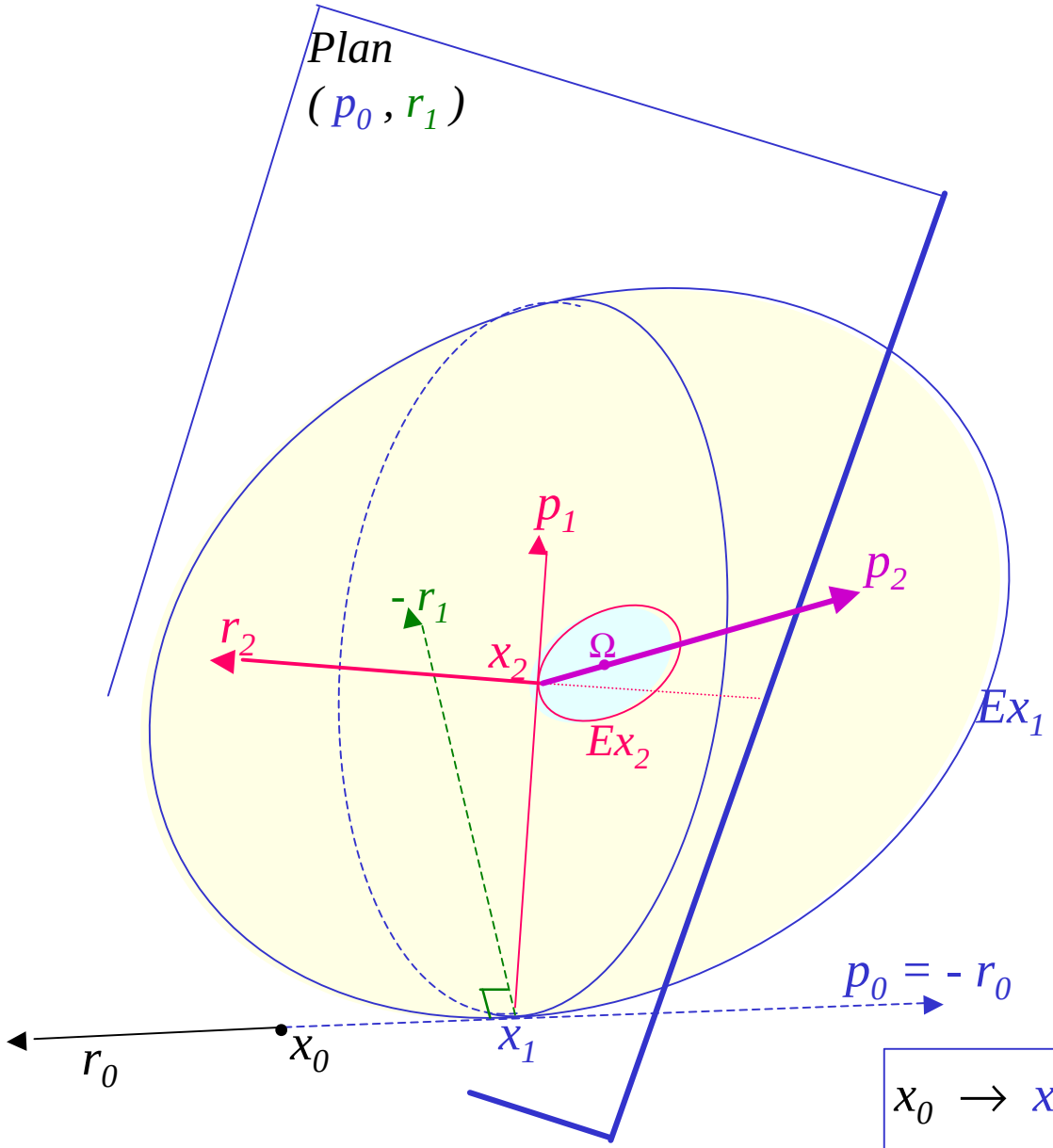
à 2- Dim

$x_0 \rightarrow x_1 \rightarrow x_2$

$\ldots r_2 = 0$

car $r_k \perp r_{k-j}$ $(j>0)$

{voir ci-après}



(b)

à 3- Dim

$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 = x_W$

… $r_3 = 0$

car $r_k \perp r_{k-j}$ $(j>0)$

{voir ci-après}

Méthode <u>semi-itérative</u> : *n* étapes max. (en principe)

. x_0

. $r_0 = C\,x_0 - d$

. $p_0 = -r_0$ et $t_0 = r_0^T r_0 / (r_0^T C\, r_0) \Rightarrow x_1 = x_0 + t_0\, p_0$

$r_1 = C\,x_1 - d$

. $p_1 = -r_1 + a_1\, p_0$ …… $p_k = -r_k + a_k\, p_{k-1}$ ……..(i)

avec $p_1^T C\, p_0 = 0$ (relation de conjugaison / C)

$\Rightarrow a_1 = r_1^T C\, p_0 / (p_0^T C\, p_0)$ …… $a_k = r_k^T C\, p_{k-1} / (p_{k-1}^T C\, p_{k-1})$…(ii)

. le minimum, dans le plan (r_1 , p_1) , est obtenu pour :

(r_k , p_k)

$t_1 = - r_1^T p_1 / (p_1^T C\, p_1)$ …… $t_k = - r_k^T p_k / (p_k^T C\, p_k)$……(iii)

(cf. éq. **(3)** *avec indice « 1 » au lieu de « 0 »)*

Propriétés :

• $r_{k+1} = C\,x_{k+1} - d = = C\,(x_k + t_k\, p_k) - d \Rightarrow r_{k+1} = r_k + t_k\, C\, p_k$ ……(iv)

$\Rightarrow p_k^T r_{k+1} = p_k^T r_k + t_k\, p_k^T C\, p_k = 0$ …… *d 'après* (iii)

Donc $\boldsymbol{r_{k+1} \wedge p_k}$

Propriétés (*suite*) :

• De $r_{k+1} = r_k + t_k\, C\, p_k$, on déduit :

$$p_{k-1}{}^T\, r_{k+1} = \underbrace{p_{k-1}{}^T\, r_k}_{=0} + t_k\, \underbrace{p_{k-1}{}^T\, C\, p_k}_{\text{conjugaison} \Rightarrow\ =0} = 0$$

Donc $\boldsymbol{r_{k+1} \wedge p_{k-1}}$

• $r_{k+1} \perp p_k$
  $r_{k+1} \perp p_{k-1}$ } donc $\boldsymbol{r_{k+1} \wedge r_k}$, d 'après (i)

• De même :

$r_{k+1} \perp r_{k-1}$
$r_{k+1} \perp r_{k-2}$
...

→ $\{r_0 , r_1 , r_2 , \ldots r_{n-1}\}$ indépendants
… et donc $r_n = 0$ (*)
$\Leftrightarrow\ x_n = x_W$

⇨ **NOMBRE FINI D ' ETAPES**

(*) *Plus généralement : $r_{rang\ C} = 0$*

Autres expressions de t_k et a_k ($k > 0$)

- $t_k = - r_k^T p_k / (p_k^T C p_k) = - r_k^T (-r_k + a_k p_{k-1}) / (p_k^T C p_k)$

Donc $t_k = r_k^T r_k / (p_k^T C p_k)$

- $a_k = r_k^T C p_{k-1} / (p_{k-1}^T C p_{k-1}) = r_k^T (1/ t_{k-1}) (r_k - r_{k-1}) / [p_{k-1}^T (1/ t_{k-1}) (r_k - r_{k-1})]$

… *d 'après* (iv)

Donc $a_k = - r_k^T r_k / (p_{k-1}^T r_{k-1})$

Une autre expression peut aussi être utilisée .
En effet, de $p_k = -r_k + a_k p_{k-1}$ *on déduit :*

$$r_k^T p_k = - r_k^T r_k + a_k \underbrace{r_k^T p_{k-1}}_{=0}$$

i.e. $r_k^T p_k = - r_k^T r_k$

D 'où $a_k = r_k^T r_k / (r_{k-1}^T r_{k-1})$

Résumé : C.G.

- x_0
- $r_0 = C\, x_0 - d$
- $p_0 = - r_0$

$k > 0$:

$$a_k = r_k^T r_k / (r_{k-1}^T r_{k-1})$$

$$p_k = -r_k + a_k\, p_{k-1}$$

$k \geq 0$:

$$t_k = r_k^T r_k / (p_k^T C p_k)$$

$$x_{k+1} = x_k + t_k p_k$$

$$r_{k+1} = r_k + t_k C p_k$$

$h_k = C p_k$

Algorithme C.G. - matrice C s.d.p.

- I-O : C (I, .) : matrice s.d.p.
  d (I, .) : vecteur second membre
  #iter (I, .) : nombre max. d 'itérations
  xi (I,O) : x_0 solution initiale , x_i solution à l 'étape [i]
  rtr(. , O) : carré de la norme des résidus

- Algo : r0 = C • xi - d : $r_0 = C\, x_0 - d$
  rtr = $r0^T$ • r0
  p0 = - r0

! *Itérations*

do i= 0 → #iter
if i > 0 then
(rtr)old = rtr
rtr = ri^T • ri
αi = rtr / (rtr)old
pi = - ri + αi . p[i-1]
endif
hi = C • pi
ti = rtr / (pi^T • hi)
x[i+1] = xi + ti . pi : $x_{i+1} = x_i + t_i\, p_i$ → *test de convergence*
(peut nécessiter un vecteur supp. x[i+1])
r[i+1] = ri + ti . hi
end do i

remplaçable par un test de convergence (e donné)

#iter < n

L 'avantage est évidemment d 'avoir un nombre effectif d 'itérations, $\mathbf{n_{iter}}$*, très inférieur à* **n** *, car le nombre d 'opérations est de l 'ordre de* $\mathbf{n^2 \cdot n_{iter}}$ **…**

4. ADAPTATION DE C.G. AUX EQUATIONS D 'OBSERVATION
(résolution par moindres carrés)

Equations d 'observation : $A\,x - b = v$ *(v : vecteur des résidus)*

\+ condition : $v^T v$ *minimum*

Þ Equations normales : $A^T A\,x - A^T b = A^T v = 0$

(vecteur résidu orthogonal aux vecteurs-colonnes de A)

Donc : $A^T A\,x = A^T b$

ou : $\boldsymbol{C\ x = d}$

N.B.

$A \leftarrow \underbrace{\sqrt{\Pi}}_{\text{poids}}\, A$

Dans le cadre C.G. on a :

- $r_0 = C\,x_0 - d = (A^T A)\,x_0 - A^T b = A^T(A\,x_0 - b) = A^T v_0$
- $r_{k+1} = A^T v_{k+1}$ (démonstration identique)
- $v_{k+1} = A\,x_{k+1} - b = A(x_k + t_k\,p_k) - b$
  $= A\,x_k - b + t_k A\,p_k = v_k + t_k A\,p_k$
- $p_k^T C\,p_k = p_k^T A^T A\,p_k = (Ap_k)^T\ (Ap_k)$

Ceci permet d 'appliquer la méthode C.G. directement aux équations d 'observation
sans avoir à calculer la matrice normale !

Algorithme C.G. pour équations d 'observation

$k = 0$ ……..

$$x_0 \rightarrow v_0 = A\,x_0 - b$$
$$r_0 = A^T v_0$$
$$p_0 = - r_0$$

$k > 0$ ……..

$$a_k = r_k^T r_k / (r_{k-1}^T r_{k-1})$$
$$p_k = - r_k + a_k\, p_{k-1}$$

$k \geq 0$ ……..

$$t_k = r_k^T r_k / (Ap_k)^T (Ap_k)$$
$$x_{k+1} = x_k + t_k\, p_k$$
$$v_{k+1} = v_k + t_k (Ap_k)$$
$$r_{k+1} = A^T v_{k+1}$$

⇩

- *on évite le calcul de* $C = A^T A$

- *2 produits Matrice ∗ vecteur :* Ap_k *et* $A^T v_{k+1}$ *à chaque itération*

  ... mais nécessite de stocker A, ou de recalculer chaque ligne (pour chaque observation) à chaque itération.

N.B. $|\,v_k\,|$ *décroît de manière monotone. En effet :*

$$v_{k+1}^T v_{k+1} = (v_k + t_k Ap_k)^T (v_k + t_k Ap_k)$$
$$= v_k^T v_k + 2\, t_k\, v_k^T A\, p_k + t_k^2 (Ap_k)^T (Ap_k)$$

Or : $v_k = v_{k+1} - t_k\, A\, p_k$ *et* $r_{k+1}^T p_k = 0$

® $|\,v_{k+1}|^2 = |\,v_k|^2 - t_k^2 (Ap_k)^T (Ap_k) + 2\, t_k (\underbrace{v_{k+1}^T A}_{(r_{k+1})^T \rightarrow \text{terme nul}})\, p_k$ ……. (QED)

III - APPLICATIONS : cadre GINS-DYNAMO et modèles de géopotentiel

1. RAPPELS

C_{lm} , S_{lm} normalisés ; $0 \le m \le l$; $S_{l0} = 0$.

Ordonnancements usuels des indices l (degré) et m (ordre) :

(i) $l = 2$ à L , $m=0$ à l ……. , C , S , C , S , etc…

(ii) $m = 0$ à L , $l = sup(m,2)$ à L, ou $l_1(m)$ à $l_2(m) \le$ L

……. , C , S , C , S , etc…

(iii) . tous les C_{lm} : $m = 0$ à L { l pair : $l_1(m)$ à $l_2(m)$, pas = 2 ; l impair : $l_1(m)$ à $l_2(m)$, pas = 2

. tous les S_{lm} : $m = 1$ à L { l pair ; l impair } comme pour les C_{lm}

Avantage de (iii) : si le système normal, $C\,x = d$, résulte :

DYNAMO--L3 → - d 'équations d 'observations pour une grille géographique régulière en *lat-long* (ex: grille de données de surface, de valeurs de *F(pot.)* sur une sphère externe - cf. méthode « *space-wise* ») , et ceci avec pondération Π(*lat*) ,

MANEGE → - ou d 'équations d 'observations régulièrement réparties sur une orbite répétitive, alors C est bloc-diagonale.

Une telle matrice C ayant cette structure sera notée N

(N comme « normale », et comme Neumann qui découvrit cette forme)

Structure de la matrice normale N

p blocs avec *l* pair

i blocs avec *l* impair

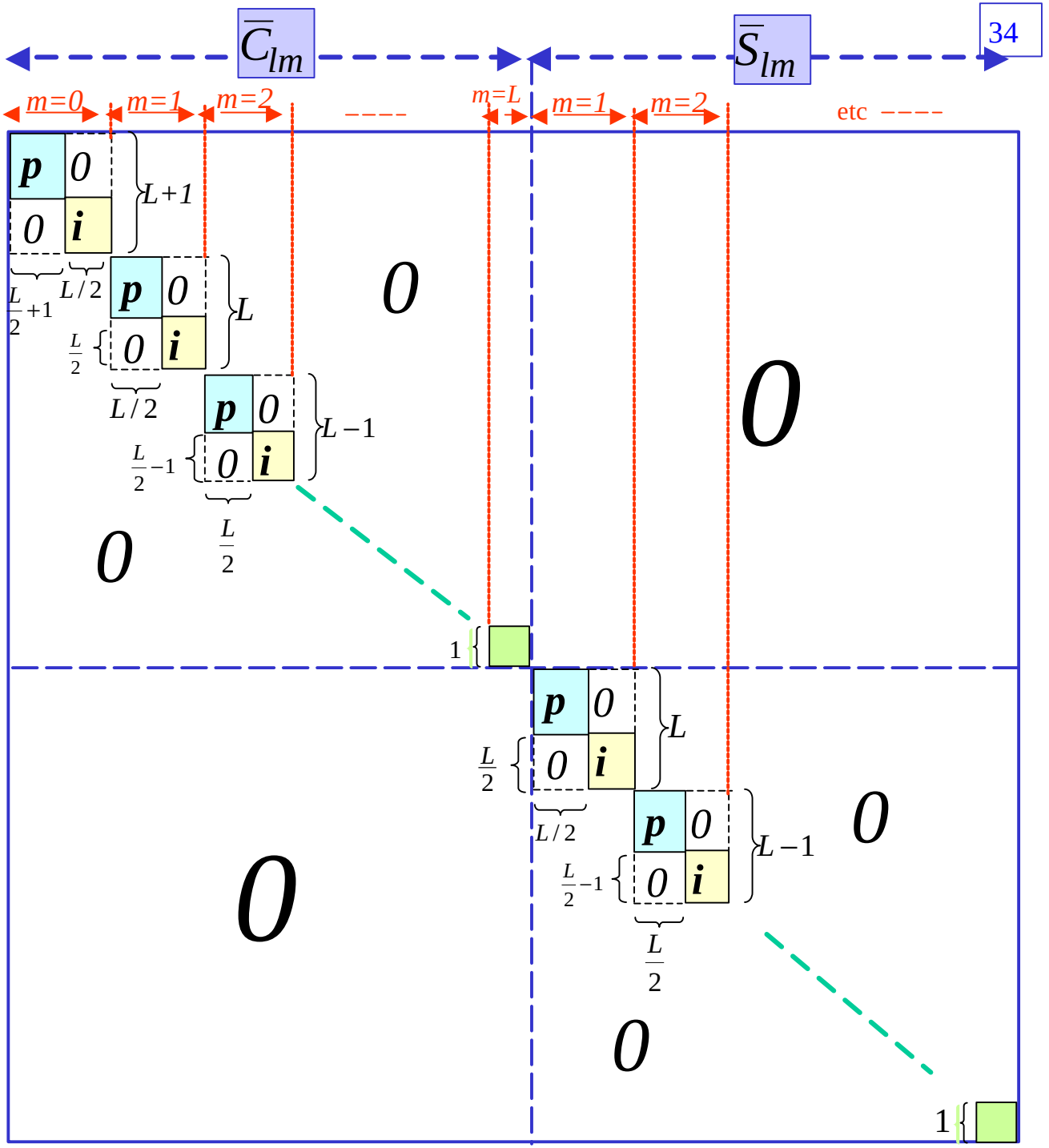
$$\mathrm{N}^{CC}_{l_1 l_2 m} = \mathrm{N}(\overline{C}_{l_1 m} \mid \overline{C}_{l_2 m})$$

(l_1 , l_2 : même parité)

$$\mathrm{N}^{SS}_{l_1 l_2 m} = \mathrm{N}(\overline{S}_{l_1 m} \mid \overline{S}_{l_2 m})$$

(id.)

Chaque (sous-)bloc est évidemment symétrique



Applications

Hypothèses

. on adopte le mode de rangement (iii)

. les inconnues autres que les (C_{lm}, S_{lm}) sont en début ou en fin du vecteur x … ou bien l 'on a réduit le(s) système(s) pour n 'avoir que les (C_{lm}, S_{lm}) comme inconnues

. on écrit $C = N + N^*$, avec N : bloc-diagonale

Conclusion

On peut se servir de N pour :

. un algorithme de Jacobi par blocs

. préconditionner le système afin d 'améliorer la convergence de tout processus (semi-) itératif.

2. METHODE DE JACOBI PAR BLOCS (BJ)

Avec les notations du chapitre I de cette seconde partie, on prend :

$$A = C \quad , \quad b = d$$

$$M = N$$

$$M\text{-}A = N\text{-}C = -N^*$$

d 'où le processus :

$$N x_{k+1} = -N^* x_k + d \quad \Leftrightarrow \quad \boxed{N (x_{k+1} - x_k) = -r_k}$$

Avantages : . en général on a $\| N^* \| << \| N \|$, d 'où CV rapide.

. N étant bloc-diagonale, résolution facile des systèmes intermédiaires

(de plus, chaque bloc est s.d.p. → triangulation par Cholesky)

Algorithme B.J. (Jacobi par blocs)

- I-O : C (I, .) : matrice s.d.p.
  d (I, .) : vecteur second membre
  N (I, .) : partie bloc-diagonale de C
  #iter (I, .) : nombre max. d 'itérations
  xi (I,O) : x_0 solution initiale (par ex. $x_0 = 0.$) ,
  x_i solution à l 'étape [i]
  rtr(. , O) : carré de la norme des résidus

- Algo : r0 = C • x0 - d

! *Itérations*

do i= 0 → #iter

rtr = ri^T • ri

dxi = Solution de [N. dxi = - ri] …. calcul de la solution à chaque fois, ou calcul de N^{-1} au début, puis produits N^{-1}.(-ri) ensuite

x[i+1]= xi + dxi : $x_{i+1} = x_i + D\, x_i$ → *test de convergence*

(peut nécessiter un vecteur supp. x[i+1])

r[i+1] = ri + C • dxi

end do i

remplaçable par un test de convergence (e donné)

#iter < n

*Comme pour l 'algorithme C.G., un nombre effectif d 'itérations, $\mathbf{n_{iter}}$, très inférieur à **n** , doit être capable de résoudre le système avec la précision voulue*

3. GRADIENTS CONJUGUES PRECONDITIONNES (PCG)

Théorème | *Le rayon spectral,* r_{CG} , *de la méthode CG est :*

$r_{CG} = [(1- \sqrt{\eta}) / (1+ \sqrt{\eta})]^2$

avec $\eta = |\lambda_{max} / \lambda_{min}|$; λ_i : *valeurs propres de* C.

Donc si n est grand, la CV est très mauvaise

IDEE . *on remplace* $C x = d$ *par* $N^{-1} C x = N^{-1} d$, *i.e.* $\bar{C} x = \bar{d}$

avec $C = N + N^*$

bloc-diagonale suivant rangement (iii)

. *résultat :* $\bar{\eta} << \eta$ ® **CV accélérée**

en général

En effet :

Si $N \approx C$, (N sera dite représentative de C) , alors $N^{-1} . C \approx I$

$\Rightarrow \quad \lambda_{max}$ et $\lambda_{min} \approx 1. \quad \Rightarrow \quad \bar{\eta} \lesssim 1.$ et $r_{PCG} \approx 0.$!

On peut résoudre $\bar{C} x = \bar{d}$ directement avec l 'algorithme CG classique si N^{-1} est rapide à calculer et $N^{-1} C$ rapide à former (ce qui est le cas si N est bloc-diagonale), ou bien appliquer à $C x = d$ un algorithme spécial qui réalise PCG implicitement.

® *voir démonstration qui suit*

Transformation de CG en PCG implicite, et formules associées

- Décomposition (Cholesky) ..... *pour expliquer les transformations (non requise dans la pratique)*

$N = U^T U$ (U : triangle sup. , par exemple)

$N^{-1} = U^{-1} U^{-T}$, avec $U^{-T} = (U^{-1})^T$

Þ $N^{-1} C x = N^{-1} d$ s 'écrit : $U^{-1} U^{-T} C (U^{-1} U) x = U^{-1} U^{-T} d$

d 'où : $\underbrace{(U^{-T} C U^{-1})}_{\tilde{C}} \underbrace{U x}_{\tilde{x}} = \underbrace{U^{-T} d}_{\tilde{d}}$

Puisque $\bar{C} = N^{-1} C = U^{-1} U^{-T} C$, on a : $\bar{C} = U^{-1} \tilde{C} U$

Þ $\bar{C}$ et $\tilde{C}$ sont semblables

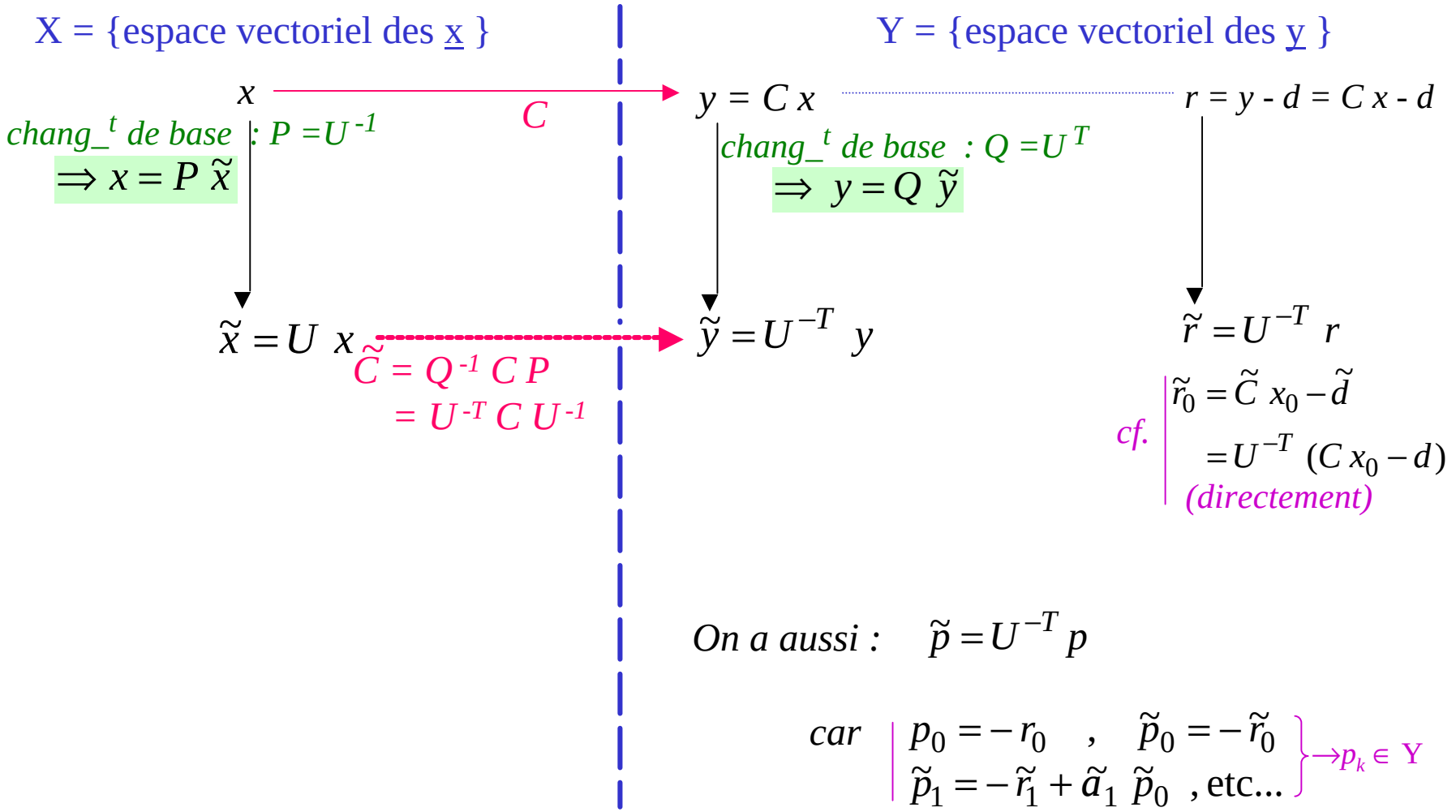
et ont donc les mêmes valeurs propres

Þ $\bar{n} = \tilde{n}$

Donc l 'algorithme portera sur $\tilde{C} \tilde{x} = \tilde{d}$

On va travailler : - dans l 'espace vectoriel des paramètres , avec les bases x et $\tilde{x}$
- dans l 'espace vectoriel des résidus , avec les bases d et $\tilde{d}$

- Transformation



On va ensuite appliquer les formules de base de C.G. au système $\tilde{C}\,\tilde{x} = \tilde{d}$,
d 'où ...

- Formules associées

(i) $x_{k+1} = U\ x_{k+1} = \tilde{x}_k + \tilde{t}_k\ \tilde{p}_k = U\ x_k + t_k\ U^{-T} p_k \quad \Rightarrow \quad x_{k+1} = x_k + \tilde{t}_k\ N^{-1} p_k$

(ii) . $\tilde{r}_{k+1} = U^{-T} r_{k+1} = \tilde{r}_k + \tilde{t}_k\ \tilde{C}\ \tilde{p}_k = U^{-T} r_k + \tilde{t}_k\ U^{-T} C \underbrace{U^{-1} U^{-T}}_{N^{-1}} p_k$

donc : $r_{k+1} = r_k + \tilde{t}_k \underbrace{U^T U^{-T}}_{Id} C\ N^{-1} p_k \quad \Rightarrow \quad r_{k+1} = r_k + \tilde{t}_k\ C\ N^{-1} p_k$

. $\tilde{p}_k = -\tilde{r}_k + \tilde{a}_k\ \tilde{p}_{k-1}$ ..... $\Rightarrow \quad p_k = -r_k + \tilde{a}_k\ p_{k-1}$

(iii) . $\tilde{r}_k^T\ \tilde{r}_k = r_k^T \underbrace{U^{-1} U^{-T}}_{N^{-1}} r_k$ ..... $\Rightarrow \quad \tilde{r}_k^T\ \tilde{r}_k = r_k^T\ r_k$

avec : $r_k = N^{-1}\ r_k$ ●

$\tilde{p}_k^T\ \tilde{C}\ \tilde{p}_k = p_k^T \underbrace{U^{-1} \overbrace{U^{-T} C U^{-1}}^{\tilde{C}} U^{-T}}_{N^{-1}\ \ N^{-1}} p_k$ ..... On pose : $\Pi_k = N^{-1} p_k$ ●

$= \Pi_k^T\ (C \Pi_k) = \Pi_k^T\ \underline{h}_k$ ..... On pose : $\underline{h}_k = C\ \Pi_k$ ●

(iv) . $\Pi_0 = N^{-1} p_0 = -N^{-1} r_0 = -\ r_0$ ..... $\Rightarrow \quad \Pi_0 = -r_0$

$\Pi_k = N^{-1}(-r_k + \tilde{a}_k\ p_{k-1})$ ..... $\Rightarrow \quad \Pi_k = -r_k + \tilde{a}_k\ \Pi_{k-1}$

$\Rightarrow$

$\tilde{t}_k = \tilde{r}_k^T\ \tilde{r}_k / (\tilde{p}_k^T\ \tilde{C}\ \tilde{p}_k) = r_k^T\ r_k / (\Pi_k^T\ \underline{h}_k)$ ●

$\tilde{a}_k = \tilde{r}_k^T\ \tilde{r}_k / (\tilde{r}_{k-1}^T\ \tilde{r}_{k-1}) = r_k^T\ r_k / (r_{k-1}^T\ r_{k-1})$ ●

Algorithme de PCG

- I-O : C (I, .) : matrice s.d.p.
  d (I, .) : vecteur second membre
  N (I, .) : matrice bloc-diagonale, ou matrice générale de préconditionnement
  #iter (I, .) : nombre max. d 'itérations
  xi (I,O) : x_0 solution initiale, x0 (par ex. $x_0 = 0.$) ; x_i solution à l 'étape [i]
  rtρ(. , O) : carré de la norme des résidus

remplaçable par un test de convergence (e donné)

- Algo : r0 = C • x0 - d ==> ρ0= Solution de { N • ρ0 = r0 }*
  rtρ= r_0^T • ρ0
  Π0= - ρ0

! *Itérations*

do i= 0 → #iter #iter < n
  if i > 0 then
    (rtρ)old = rtρ
    rtρ = ri^T • ρi
    $\tilde{\alpha}$i = rtρ / (rtρ)old
    Πi = -ρi + $\tilde{\alpha}$i . Π [i-1]
  endif
  $\underline{h}$i = C • Πi
  $\tilde{t}$i = rtρ / (Πi^T • $\underline{h}$i)
  x[i+1] = xi + $\tilde{t}$i . Πi → *test de convergence*
  r[i+1] = ri + $\tilde{t}$i . $\underline{h}$i
  ρ[i+1]= Solution de { N • ρ[i+1] = r[i+1] }*
end do i

≡ Algo. CG si $N=N^{-1}= I$ car alors :

$r_k = r_k$
$h_k = \underline{h}_k$
$p_k = P_k$
$a_k = \tilde{a}_k$
$t_k = \tilde{t}_k$

* Il peut être avantageux, si N^{-1} est très rapide à calculer et/ou si #iter est grand, de pré-calculer N^{-1} et d'effectuer les produits $N^{-1} r_0$, $N^{-1} r_1$, $N^{-1} r_2$, ... Si N^{-1} n'existe pas, choisir un algorithme de résolution de $N r = r$ qui soit de norme minimum (par ex., "Cholesky étendu").

4. ADAPTATIONS DE "BJ" ET DE "PCG" POUR LE TRAITEMENT DIRECT DES EQUATIONS D'OBSERVATION.

D'après II.4 , on doit faire les changements suivants :

$$C\,x = d \quad \rightarrow \quad A\,x - b = v$$

$$r_0 = C\,x_0 - d \quad \rightarrow \quad \begin{cases} v_0 = A\,x_0 - b \\ r_0 = A^T v_0 \end{cases}$$

- "BJ" modifié

$$r_{i+1} = r_i + C\,Dx_i \quad \rightarrow \quad \begin{cases} v_{i+1} = v_i + A\,Dx_i \\ r_{i+1} = A^T v_{i+1} \end{cases}$$

- "PCG" modifié

$$\underline{h}_i = C\,P_i \quad \rightarrow \quad \begin{cases} g_i = A\,P_i \\ \underline{h}_i = A^T g_i \end{cases}$$

$$\tilde{t}_i = r^T r \,/\, (P_i^T\,\underline{h}_i) \quad \rightarrow \quad \tilde{t}_i = r^T r \,/\, (g_i^T\,g_i)$$

$$x_{i+1}\,,\, r_{i+1} \quad \rightarrow \quad \begin{cases} \text{idem} \\ v_{i+1} = v_i + \tilde{t}_i\,g_i \end{cases}$$

... et utilisation de N

5. METHODES "BJ-Comb" et "PCG-Comb" dans un contexte de parallélisation.

Soit J + K groupes d'observations :

J groupes d'éq. d'observation

$$\begin{cases} A_1 x - b_1 = v_1 \\ \vdots \\ A_J x - b_J = v_J \end{cases}$$

K groupes d' éq. normales

$$\begin{cases} C_1 x - d_1 = 0 \\ \vdots \\ C_K x - d_K = 0 \end{cases}$$

$$\Leftrightarrow \quad \mathbf{C}\, x = \boldsymbol{d} \quad \text{avec} \quad \begin{cases} \mathbf{C} = \sum_{u=1}^{J} A_u^T A_u + \sum_{w=1}^{K} C_w \\ \boldsymbol{d} = \sum_{u=1}^{J} A_u^T b_u + \sum_{w=1}^{K} d_w \end{cases}$$

→ DYNAMO-C

On suppose toujours que :

- le rangement des (C_{lm} , S_{lm}) est de type (iii)
- $\mathbf{C} = N + N^*$ avec N : bloc-diagonale, ou bien N est une matrice (simple à inverser) "représentative" de $\mathbf{C}$, par ex :

$$N = \tilde{\mathbf{C}} = \sum_{u} A_u^T A_u + \sum_{w} C_w \quad \text{, avec } card\{u\} << J \quad card\{w\} << K$$

⇒ Possibilité de rechercher la solution pour ces systèmes mixtes, directement et en utilisant J+K processeurs

"BJ-Comb" ⟷ "PCG-Comb"

cf. séquences { ... }//

Algorithme "B.J.- Comb"

- I-O : J, K (I, .) : nombres de groupes de matrices A et C, respectivement
  A_u (I, .) : J matrices d'observation
  b_u (I, .) : J vecteurs seconds membres des équations d'observation
  C_w (I, .) : K matrices normales
  d_w (I, .) : K vecteurs seconds membres des équations normales
  N (I, .) : partie "représentative" de C , peut être bloc-diagonale
  #iter (I, .) : nombre max. d 'itérations
  xi (I,O) : x_0 solution initiale (par ex. $x_0 = 0.$) ,
  x_i solution à l 'étape [i]
  rtr(. , O) : carré de la norme des résidus = $r^T r$, pour les J+K groupes

remplaçable par un test de convergence (e donné)

- Algo : r0 (= r_0) : $\left\{ \begin{array}{l} v_{u,0} = A_u \bullet x_0 - b_u \\ r_{u,0} = A_u^T \bullet v_{u,0} \end{array} \right.$ // $u = 1,\dots J$ ---- $\left\{ r^n_{w,0} = C_w \bullet x_0 - d_w \right.$ // $w = 1,\dots K$

! *Itérations*

do i= 0 → #iter

rtr = $ri^T \bullet ri$: = *somme sur les* $r_{u,i}$ *et* $r_{w,i}$ *(u= 1 à J ; w=1 à K)*

dxi = Solution de [N. dxi = - ri] …. calcul de la solution à chaque fois, ou calcul de N^{-1} au début, puis produits $N^{-1}.(-ri)$ ensuite

x[i+1]= xi + dxi : $x_{i+1} = x_i + D\, x_i$ → *test de convergence*

r[i+1] : $\left\{ \begin{array}{l} v_{u,\,i+1} = v_{u,i} + A_u \bullet dx_i \\ r_{u,\,i+1} = A_u^T \bullet v_{u,\,i+1} \end{array} \right.$ // $u = 1,\dots J$ ---- $\left\{ r^n_{w,\,i+1} = r^n_{w,i} + C \bullet dxi \right.$ // $w = 1,\dots K$

end do i

Remarque *: notation* r^n *pour les résidus des systèmes normaux, pour les distinguer (à cause de l'indiçage) de ceux des autres systèmes normaux (non formés) associés aux équations d'observation.*

Algorithme "P.C.G.- Comb"

- I-O :

J, K (I, .) : nombres de groupes de matrices A et C, respectivement
A_u (I, .) : J matrices d'observation
b_u (I, .) : J vecteurs seconds membres des équations d'observation
C_w (I, .) : K matrices normales
d_w (I, .) : K vecteurs seconds membres des équations normales
N (I, .) : partie "représentative" de C , peut être bloc-diagonale
#iter (I, .) : nombre max. d 'itérations
xi (I,O) : x_0 solution initiale (par ex. $x_0 = 0.$) ,
x_i solution à l 'étape [i]

idem "BJ-Comb"

.....

rtρ(. , O) : carré de la norme des résidus d'ajustement ,
= résidus des équations "normales" (réelles ou virtuelles)
pour les J+K groupes

$v_{u,i}$: résidus des J groupes d'observations

Remarque : *Tout comme pour "B.J.-Comb", nous noterons* r^n *les résidus des systèmes normaux, pour les distinguer (à cause de l' indiçage) de ceux des autres systèmes normaux (non formés) correspondant aux systèmes d'équations d'observation.*

- Algo : → *planche suivante* **...**

Algorithme "P.C.G.- Comb" (*suite et fin*)

• Algo : r0 :

$$\begin{cases} v_{u,0} = A_u \bullet x_0 - b_u \\ r_{u,0} = A_u^T \bullet v_{u,0} \end{cases}_{u=1,\dots J}^{//} \qquad \left\{ r^n_{w,0} = C_w \bullet x_0 - d_w \right\}_{w=1,\dots K}^{//}$$

$$\rightarrow \quad \left| \begin{array}{l} \rho 0 = \text{solution de } \{ N\, \rho 0 = r0 \}\ * \\ r t\rho = r0^T \bullet \rho 0 \\ \Pi 0 = -\rho 0 \end{array} \right.$$

! *Itérations*

do i= 0 → #iter

if i > 0 then

$(r t\rho)old = r t\rho$

$r t\rho = r_i^T \bullet \rho_i$ ⟶ *somme sur les* $r_{u,i}$ *et* $r_{w,i}$ *(u= 1 à J ; w=1 à K)*

$\bar{\alpha} i = r t\rho / (r t\rho)old$

$\Pi i = -\rho i + \bar{\alpha} i \,.\, \Pi [i-1]$

endif

$$\begin{cases} g_{u,i} = A_u \bullet \Pi i \\ h_{u,i} = A_u^T \bullet g_{u,i} \end{cases}_{u=1,\dots J}^{//} \qquad \left\{ \underline{h}_{w,i} = C_w \bullet \Pi i \right\}_{w=1,\dots K}^{//}$$

$$\bar{t} i = r t\rho \Big/ \left(\sum_{u=1}^{u=J} g_{u,i}^T \bullet g_{u,i} + \sum_{w=1}^{w=K} \Pi i^T \bullet \underline{h}_{w,i} \right)$$

$x[i+1] = xi + \bar{t} i \,.\, \Pi i$ → *test de convergence*

r[i+1] :

$$\begin{cases} v_{u,\,i+1} = v_{u,i} + \bar{t}_i\, g_{u,i} \\ r_{u,\,i+1} = r_{u,i} + \bar{t}_i\, h_{u,i} \end{cases}_{u=1,\dots J}^{//} \qquad \left\{ r^n_{w,\,i+1} = r^n_{w,i} + \bar{t}_i\, \underline{h}_{w,i} \right\}_{w=1,\dots K}$$

$\rho[i+1]$= Solution de $\{ N \bullet \rho[i+1] = r[i+1] \}$*

end do i

*calcul de la solution à chaque fois, ou calcul de N^{-1} au début, puis produits $N^{-1} \bullet r_i$ ensuite

Fin de l'algorithme

Quelques remarques en conclusion

❶ Systèmes singuliers et gradients conjugués (CG)

Soit le problème : $C\,x = d$

avec : $x^T x$: minimum

==> On minimise la fonctionnelle : $\Theta\,(x\,,\,\lambda\,) = x^T x \;-\; 2\,\lambda^T\,(\,Cx - d\,)$

multiplicateurs de Lagrange

$\partial\,\Theta / \partial x = 0 \;\Rightarrow\; x = C^T\,\lambda$

$\Rightarrow\; x \in$ espace vect. {lignes de C }

N.B. Si $C = A^T A$, $x = A^T \underbrace{A\,\lambda}_{\mu} = A^T\,\mu$ $\Rightarrow\; x \in$ espace vect. {lignes de A }

... et r_k également.

C.G. fournit automatiquement la solution de norme minimum
du problème de moindres carrés
(donc on obtient cette solution même si C^{-1} n'existe pas)

❷ Stabilisation par des lois de type "Kaula" :

- application habituelle (chaîne DYNAMO) sur les systèmes normaux

- peut aussi se faire sur les équations d'observation en écrivant :

$$(\Delta C_{lm} , \Delta S_{lm}) = 0 \pm \alpha / l^2$$

❸ Obtention des variances et des covariances de la solution :

- problèmes de potentiel gravitationnel (avec harmoniques sphériques) :
  une estimation satisfaisante peut être obtenue à partir de l'inverse de la matrice de préconditionnement N (bloc-diagonale) , l'inverse exacte de C n'étant calculée - si c'est possible, que pour la solution finale.

- problème quelconque,
  ... ou nécessité d'avoir la matrice de covariance exacte :
  la possibilité dépend de la puissance de calcul !!!

ET VOILA LA SOLUTION !


FIN
Emmanuel Mallart, cidehom.com