Chapitre II

Algèbre linéaire

I - Méthodes de résolution de systèmes linéaires
II - Algèbre linéaire et perturbations
III - Méthodes itératives
IV - Méthodes de descente
V - Préconditionnement.

I

* On cherche à résoudre le pb suivant :

$Ax = b$. La première idée est d'inverser la matrice. Mais ceci est un pb difficile.

* Cas où A est diagonale. La solut° est immédiate : $A^{-1} = D\left(\frac{1}{a_{11}}, \frac{1}{a_{22}} \dots \frac{1}{a_{nn}}\right)$

La solution est donc :

$$x_i = \frac{1}{a_{ii}} * b_i$$

* Cas où A est triangulaire supérieure.

$$x_n \leftarrow \frac{b_n}{a_{nn}}$$

Pour $i \leftarrow n-1$ à 1 pas -1 faire

$$x_i \leftarrow \left(b_i - \sum_{j=i+1} a_{ij} x_j\right) / a_{ii}$$

Fin Pour

* Méthodes directes.

L'idée est de remplacer le système $Ax = b$ ds lequel la matrice A est pleine, par plusieurs systèmes faciles à résoudre car creux.
L'idéal serait de remplacer A par PDP^{-1} avec D diagonal mais tp couteux.
L'alternative est de remplacer A par LU avec L triangulaire inf et U triangulaire sup.
On résoud alors $Ly = b$ puis $Ux = y$.

La factorisation LU, qd elle existe, n'est pas unique. Pour la rendre unique, on ajoute la contrainte que la diagonale de L est à 1. C'est la factorisation de Gauss.
Pour les matrices définies positives et symétriques, on utilise la factorisation de Cholesky $A = HH^T$ où H est triangulaire inférieure.

* Factorisation de Gauss.

→ On applique la méthode du pivot de Gauss (=> pas de 0 sur la diagonale)

$$A^{(1)} = \begin{pmatrix} a_{11} & & a_{1n} \\ & & \\ a_{n1} & & a_{nn} \end{pmatrix} ; \Rightarrow M^{(1)} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 \cdots & 0 \\ \vdots & 0 & & \\ -\frac{a_{n1}}{a_{11}} & 0 & & 1 \end{bmatrix}$$

$$\Rightarrow L^{(1)} = M^{(1)-1} = \begin{bmatrix} 1 & 0 & \cdots & & 0 \\ \frac{a_{21}}{a_{11}} & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ \frac{a_{n1}}{a_{11}} & 0 & \cdots & & 1 \end{bmatrix}$$

$$\Rightarrow L^{(1)} = -M^{(1)} + 2I$$

et $A^{(2)} = M^{(1)} A^{(1)}$ (ça revient à faire Gauss)

On réitère l'opération $(n-1)$ fois.

A la fin on a $L = L^{(1)} L^{(2)} L^{(3)} \dots L^{(n-1)}$
et $U = A^{(n)}$

et $A = LU$

Rq : $L = L^{(1)} + L^{(2)} \dots + L^{(n-1)} - I^{(n-2)}$

Coût : $\frac{n^3}{3}$ additions / multiplications et $\frac{n(n-1)}{2}$ divisions.

$\rightarrow$ $\det A = \det LU = \det L \cdot \det U = \prod l_{ii} \cdot \prod u_{ii}$
$= \prod u_{ii}$

Algo :

```
Pour j <- 1 à n-1
    Pour i <- j+1 à n
        A(i, j) = A(i,j)/A jj
        Pour k <- j+1 à n
            A(i, k) = A(i, k) - A(i, j) * A(j, k)
        Fin Pour
    FinPour
FinPour
```

* Une matrice A admet une factorisation LU ssi tous les blocs A_p, $1 \leq p \leq n$ sont inversibles.

* <u>Factorisation de Cholesky</u>

<u>Algo</u> : $L(1,1) \leftarrow \sqrt{A(1,1)}$

Pour $i \leftarrow 2$ à n

Pour $j \leftarrow 1$ à $i-1$

$$L(i,j) \leftarrow \frac{1}{L(j,j)}\left(A(i,j) - \sum_{k=1}^{j-1} L(i,k) * L(j,k)\right)$$

Fin Pour

$$L(i,i) \leftarrow \left[A(i,i) - \sum_{k=1}^{i-1} L^2(i,k)\right]^{1/2}$$

<u>Coût</u> : $\frac{n^3}{6}$ addit°/multiplications.

$\frac{n(n-1)}{2}$ divisions.

n évaluations.

* La méthode de Gauss peut-être bloquée si on tombe sur un pivot nul. La méthode de Gauss avec pivotage partiel résoude ce pb. en échangeant les lignes du pb lorsque le pivot est nul. On utilisat° les matrices de permutations.

Ainsi, si on note P la matrice des permutations, toute matrice inversible A peut s'écrire sous la forme

$$PA = LU \quad \text{avec } P \text{ matrice de permutations.}$$

<u>Rq</u> : pivotage partiel $\Rightarrow$ meilleure stabilité.

A chaque étape, on prend comme pivot (permutat°) l'élément de la diago qui est le ⊕ grand en valeur absolue.

(II) * Normes vectorielles

- Pptés d'une norme $\|.\|$:

$$\|x\| \geq 0$$
$$\|\alpha x\| = |\alpha| . \|x\|$$
$$\|x+y\| \leq \|x\| + \|y\|$$

- Normes les plus utilisées :

1) $\|x\|_1 = \sum_{i=1}^{n} |x_i|$

2) $\|x\|_2 = \sqrt{x^T x}$

3) $\|x\|_\infty = \max |x_i|$

- Norme générale :

$$\|x\|_p = \left(\sum |x_i|^p\right)^{1/p}$$

- Inégalité de Hölder

$$\sum_{i=1}^{n} |x_i y_i| \leq \|x\|_p . \|x\|_q \quad \text{et} \quad \frac{1}{p} + \frac{1}{q} = 1$$

- Inégalité de Cauchy

$$\sum_{1}^{n} |x_i y_i| \leq \sqrt{\sum |x_j|^2} . \sqrt{\sum |y_j|^2}$$

- Tes les normes de $\mathbb{R}^n$ st équivalentes

* Norme matricielle.

- Ptés d'une norme matricielle.

  1) $\|A\| \geq 0$ et $\|A\| = 0$ ssi $A = 0$

  2) $\|\alpha A\| = |\alpha| . \|A\|$.

  3) $\|A + B\| \leq \|A\| + \|B\|$.

  Si de plus on a $\|AB\| \leq \|A\| . \|B\|$, la norme est ss-multiplicative.

- A tte norme vectorielle on peut faire correspondre une norme matricielle :

$$\|A\| = \sup_{\|x\|=1} \|Ax\| = \sup_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|}$$

  Une telle norme est dite subordonnée et notée $lub(A)$.

- Norme les plus utilisées.

  1°) somme des colonnes : $\|A\|_1 = \sup_{\|x\|_1 = 1} \|Ax\|_1 = \max_{j=1..n} \sum_i |a_{ij}|$

  2°) $\|A\|_2 = \sup_{\|x\|_2 = 1} \|Ax\|_2 = \sqrt{\rho(A^T A)}$

  3°) $\|A\|_3 = \sup_{\|x\|_\infty = 1} \|Ax\|_\infty = \max_{i=1..n} \sum_{j=1}^{n} |a_{ij}|$ somme des lignes

Une norme est consistante ssi $\|Ax\| \leq \|A\| . \|x\|$. La norme 2 (ou de Schur) est consistante avec la norme euclidienne

consistante + subordonnée => ss- multiplicative

Le rayon spectral d'une matrice est inférieur (ou égal) à la norme de celle-ci :

$$\rho(A) \leq \|A\|.$$

* Conditionnement d'une matrice.

Un pb est bien conditionné si une petite variation des entrées entraîne une petite variation des sorties.
Dans la même idée, on détermine le conditionnement d'une matrice

$$K(A) = \|A\| . \|A^{-1}\|$$

Il faut $K(A)$ le plus petit possible.

<u>Théorème</u> : Si A est régulière et si

$$\frac{\|\Delta A\|}{\|A\|} < \frac{1}{K(A)} \text{ alors } A+\Delta A \text{ est régulière.}$$

→ $\frac{1}{K(A)}$ mesure la distance qui sépare d'une matrice singulière.

<u>Justification</u> :

- Perturbation de b : $A(x+\Delta x) = b + \Delta b$

  ↳ $A\Delta x = \Delta b$

  ↳ $\|\Delta x\| = \|A^{-1}\| . \|\Delta b\|$

<u>or</u> $Ax = b$ et $\|Ax\| = \|b\| \leq \|A\| . \|x\|$

$\Rightarrow \|\Delta x\| . \|b\| \leq \|A\| . \|x\| . \|A^{-1}\| . \|\Delta b\|$

$\rightarrow \frac{\|\Delta x\|}{\|x\|} \leq K(A) . \frac{\|\Delta b\|}{\|b\|}$

- Perturbation de A : $(A+\Delta A)(x+\Delta x) = b$

  ↳ $A\Delta x = -\Delta A(x+\Delta x)$

  ↳ $\|\Delta x\| = \|A^{-1}\| . \|\Delta A\| . \|x+\Delta x\|$

$\Rightarrow \|\Delta x\| \leq \|A^{-1}\| . \|\Delta A\| . (\|x\| + \|\Delta x\|)$

$\Rightarrow \|\Delta x\| (1 - \|\Delta A\| . \|A^{-1}\|) \leq \|A^{-1}\| \|\Delta A\| . \|x\|$

Si $\|A^{-1}\| . \|\Delta A\| << 1$

$\Rightarrow \frac{\|\Delta x\|}{\|x\|} \leq K(A) . \frac{\|\Delta A\|}{\|A\|}$

• Perturbation de A et b : $(A + \Delta A)(x + \Delta x) = b + \Delta b$.

$\hookrightarrow A \Delta x + \Delta A x + \underbrace{\Delta A \Delta x}_{\text{négliger}} = \Delta b$.

$\hookrightarrow \Delta x = - A^{-1} \Delta A \cdot x + A^{-1} \Delta b$.

$$\rightarrow \frac{\|\Delta x\|}{\|x\|} \leq \|A^{-1}\| \cdot \|\Delta A\| + \|A^{-1}\| \cdot \frac{\|\Delta b\|}{\|x\|}$$

$$\Rightarrow \frac{\|\Delta x\|}{\|x\|} \leq K(A) * \frac{\|\Delta A\|}{\|A\|} + K(A) \cdot \frac{\|\Delta b\|}{\|A\| \cdot \|x\|}$$

or : $Ax = b \Rightarrow \|A\| \cdot \|x\| \geq \|b\|$

$$\Rightarrow \frac{\|\Delta x\|}{\|x\|} \leq K(A) \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right)$$

* Analyse de l'erreur active

• $m(x) = x + \eta x$ ou $m(x) = \dfrac{x}{1+\eta}$ avec $|\eta| < eps$

• On pose $s_k = x_1 + \dots + x_k$

$\hookrightarrow s_k = s_{k-1} + x_k$

* $m(s_k) = s_k + e_k = \dfrac{s_k}{1+\varepsilon_k}$; $m(x_k) = x_k (1 + \eta_k)$

$\rightarrow s_k + e_k + \varepsilon_k m(s_k) = s_{k-1} + e_{k-1} + x_k + \eta_k x_k$

$\Rightarrow |e_k| \leq |e_{k-1}| + eps |x_k| + eps |m(s_k)|$

On pose $S_k = S_{k-1} + |x_k| + |m(s_k)|$

$\Rightarrow |e_k| \leq eps \, S_k$

Analyse active de l'erreur : $|x \oplus y - m(x \oplus y)| \leq eps \, |m(x \oplus y)|$

* Erreur d'un produit vectoriel.

↳ soit $A = xy^T = (a_{ij})$ et $m(a_{ij}) = x_i y_j (1 + \eta_{ij})$

$m(A) = xy^T + \Delta$ avec $\|\Delta\| \leq eps \|xy^T\|$.

* Erreur d'un produit matriciel.

↳ soit $C = A.B$, $A \in \mathbb{R}^{m \times n}$ $B \in \mathbb{R}^{n \times p}$

Exprimons

$A = \begin{bmatrix} a_1 \\ a_m \end{bmatrix}$ avec a_i vecteur ligne.

$B = [b_1 \; ... \; b_p]$ avec b_j vecteur colonne

$\Rightarrow c_{ij} = a_i b_j$ donc $m(c_{ij}) = (a_i + \Delta a_i) b_j$

avec $\|\Delta a_i\| \leq \gamma_n \|a_i\|$

erreur en retard $\Rightarrow m(c_j) = (A + \Delta A) b_j$

avec $\begin{cases} \|\Delta A\| \leq \gamma_n \|A\|. \\ c_j \text{ vecteur colonne.} \end{cases}$

erreur en avance $\Rightarrow \|c - m(c)\| \leq \gamma_n \|A\|.\|B\|$.

* Préconditionnement d'une matrice

↳ soit $A_0 x = b$ avec A_0 mal conditionnée c-à-d $\|A_0 B_0 - I\|$ est supérieur à un seuil donné. On construit le système préconditionné $B_0 A_0 x = B_0 b$.

On pose $A_1 = B_0 A_0$

On calcul B_1 l'inverse de A_1.

Si on est tjrs au dessus du seuil, on calcule B_2 l'inverse de $A_2 = B_1 B_0 A_0$

On réitère jusqu'à passer en-dessous du seuil.

III Méthode itérative

↳ Construire une suite qui converge vers la solution.

Exemple

Une méthode itérative de la forme :

$$\begin{cases} x^{(0)} \text{ donné} \\ x^{(k+1)} = B x^{(k)} + f \end{cases}$$

Celle-ci est dite consistante si la solution x vérifie $x = Bx + f \iff f = (I - B) A^{-1} b$.

On note $e^{(k)} = x^{(k)} - x$ l'erreur de la $k^{\text{ème}}$ itération.

Théorème : Une méthode consistante converge vers la solution pour toute valeur initiale ssi $\rho(B) < 1$

Théorème : les erreurs $e^{(k)} = x^{(k)} - x$ satisfont

$$\sup_{e^{(0)} \neq 0} \lim_{k \to +\infty} \sqrt[k]{\frac{\|e^{(k)}\|}{\|e^{(0)}\|}} = \rho(B)$$

* **Méthode itérative linéaire**

↳ décomposit° de A (splitting) : $A = P - N$ avec P inversible

On pose $P x^{(k+1)} = N x^{(k)} + b$.

$$\Rightarrow x^{(k+1)} = B x^{(k)} + f \text{ avec } \begin{cases} B = P^{-1} N \\ f = P^{-1} b \end{cases}$$

On peut aussi écrire :

$$x^{(k+1)} = B x^{(k)} + P^{-1} r^{(k)}$$

avec $r^{(k)} = b - A x^{(k)}$

* Décomposition d'une matrice : Jacobi vs Gauss Siedel

$$A = \begin{bmatrix} \ddots & & U \\ & D & \\ L & & \ddots \end{bmatrix}$$

Méthode	Décomposition	$M^{-1}N$	Itération
Jacobi	$A = D - (L+U)$	$J = -D^{-1}(L+U)$	$Dx^{(k+1)} = -(L+U)x^{(k)} + b$
Gauss Siedel	$A = (D+L) - U$	$G = (D+L)^{-1}U$	$(D+L)x^{(k+1)} = Ux^{(k)} + b$

* Amélioration : méthodes relaxées.

↳ On remplace l'itéré calculé par une méthode précédente ($\hat{x}^{(k+1)}$) par $x^{(k+1)}$ en faisant intervenir $x^{(k)}$: prise en compte des plusieurs états antérieurs avec pondérat° :

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

- Jacobi relaxé

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

- Gauss - Siedel relaxé

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

* Conditions de convergence à priori

- A à diagonale strictement dominante $\Rightarrow$ convergence

- A et $2D - A$ sym. définies positives, alors convergence par Jacobi et $\rho(J) = \| J \|$

- A sym. def. positivés $\Rightarrow$ convergence par Jacobi relaxée si $0 < \omega < \frac{2}{\rho(D^{-1}A)}$

- A def. pos. sym $\Rightarrow$ Gauss-Si il converge

- Si convergence par Jacobi alors convergence par Jacobi relaxé pour $0 < \omega < 1$.

* Nbre d'itérations :

$$k_{min} \simeq - \frac{\log(\varepsilon)}{R(B)} \quad \begin{cases} \varepsilon : \text{erreur admise} \\ R(B) = - \log \rho(B) \end{cases}$$

* Test d'arrêt

1) $e^{(k+1)} = B e^{(k)} \Rightarrow \| e^{(k+1)} \| \leq \frac{\| B \|^{k+1}}{1 - \| B \|} \cdot \| x^{(1)} - x^{(0)} \|$

2) $\frac{\| P^{-1} r^{(k)} \|}{\| P^{-1} r^{(0)} \|} \leq \varepsilon$

IV

La résolution d'un système linéaire peut être vu comme la minimisation d'une fctionnelle.

$$Ax = b, \quad A \in \mathbb{R}^{n \times n}, \quad x \in \mathbb{R}^n$$

La fonctionnelle est définie comme suit :

$$\bullet \; J(x) = \frac{1}{2}\left(x^T A x\right) - x^T b.$$

Gradient → $\boxed{\nabla J(x) = \frac{1}{2}\left(A^T + A\right)x - x^T A = Ax - b.}$

Résoudre le système revient à résoudre $\nabla J(x) = 0$.

→ on a donc un pb. d'optimisation.

On cherche x^* tq $J(x^*)$ soit $\begin{cases} \text{minimum.} \\ \text{maximum.} \end{cases}$

Rappels.

gradient • $\nabla J = \left[\frac{\partial J}{\partial x_1} ; \frac{\partial J}{\partial x_2} \cdots\cdots \frac{\partial J}{\partial x_n} \right]$

hessienne • $HJ = \nabla^T J . \nabla J.$

$$= \begin{bmatrix} \frac{\partial^2 J}{\partial x_1^2} & \cdots & \frac{\partial^2 J}{\partial x_1 x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 J}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 J}{\partial x_n^2} \end{bmatrix}$$

norme de l'énergie • $\|x\|_A^2 = x^T A x$

• $A \leq B$ ssi $\|x\|_A \leq \|x\|_B$

fonction convexe : $J(\lambda x + (1-\lambda) y) \leq \lambda J(x) + (1-\lambda) J(y)$

fortement convexe : $J(v) \geq J(u) + \nabla J(u)^T (v-u) + \frac{a}{2} \| u - v \|^2$

extremum : x^* extremum $\Rightarrow \nabla J(x^*) = 0$

J convexe et $\nabla J(x^*) = 0 \Rightarrow x^*$ extremum

minimum local : $\exists B(x^*, r)$ tq $\forall x \in B$, $J(x^*) \leq J(x)$

minimum global : $\forall x$, $J(x^*) \leq J(x)$.

On appelle direction admissible en x, tout vecteur d tq $\exists h > 0$: $[x ; x + hd] \subset U$. On note $D(x)$ l'ensemble des directions admissibles.

Un vecteur d est une direction de descente pour une fonctionnelle J au pt x si $\forall \gamma, \exists h \in]0; \gamma[$ tel que $J(x + hd) \leq J(x)$.

L'idée de l'algo. est de partir d'un pt $x^{(0)}$ tq $\nabla J(x^{(0)}) \neq 0$ et de construire $x^{(1)}$ en utilisant une direction de descente depuis $x^{(0)}$.

1°) DIRECTION

→ Cauchy : $d = -\nabla J(x)$.

→ Newton : $d = -H^{-1}(J(x)) \nabla J(x)$. avec H définie positive

2°) PAS

→ Pas fixe : pas fixé, tjrs le même

→ Pas Optimal : pas choisi à chaque itérat° tq : $J(x^{(0)} + \gamma d)$ soit optimal.

* Algo du gradient à pas fixe

Etape 1 → $x^{(0)}$, on calcule $d_0 = -\nabla J(x^{(0)})^T$

↳ $x^{(1)} = x^{(0)} + h \times d_0$

Etape k → $x^{(k)}$, on calcule $d_k = -\nabla J(x^{(k)})^T$

↳ $x^{(k+1)} = x^{(k)} + h\, d_k$

Arrêt → $\|\nabla J(x^{(k)})\| \leq \varepsilon_{seuil}$

* Algo du gradient à pas optimal

Etape 1 → $x^{(0)}$, on calcule $d_0 = -\nabla J(x^{(0)})^T$

on calcule $h_0 = \arg\left[\min_{h \geq 0} J\left(x^{(0)} - h \times \nabla J(x^{(0)})^T\right)\right]$

rappel : $z = |z| e^{i.\arg(z)}$

↳ $x^{(1)} = x^{(0)} + h_0 * d_0$

Etape k → $x^{(k)}$, on calcule $d_k = -\nabla J(x^{(0)})^T$

on calcule $h_k = \arg\left[\min_{h \geq 0} J\left(x^{(0)} - h * \nabla J(x^{(0)})^T\right)\right]$

Arrêt → $\|\nabla J(x^{(k+1)})\| \leq \varepsilon_{seuil}$

En général, $\Theta = 10^{-6} * \|\nabla J(x^{(0)})\|$

⚠ Vérifier que la solution n'est pas un pt selle.

Théorème : Si $J(x) = \frac{1}{2} x^T A x - b^T x$ avec A sym. déf. positive, la méthode du gradient à pas optimal cvge vers l'optimum unique $x^* = A^{-1}b$.
De plus $\| x^{(k+1)} - x^* \| < \| x^{(k+1)} - x^* \|_A^2 \left(\frac{K(A)-1}{K(A)+1} \right)^2$
où $K(A)$ est le conditionnement de A relativement à la norme 2.

* Algo de Newton

Soit $x^{(0)} \in U$. Par Taylor, on a au voisinage de $x^{(0)}$:

$$J^{(0)}(x) = J(x^{(0)}) + \nabla J(x^{(0)})[x - x^{(0)}] + \frac{1}{2}(x - x^{(0)})^T J''(x^{(0)})(x - x^{(0)})$$

On minimise la fct° quadratique J^0 et on obtient $x^{(1)}$:

$$x^{(1)} = x^{(0)} - [H(J(x^{(0)}))]^{-1} \nabla J(x^{(0)})^T$$

A l'étape k, on construit $J^{(k)}$ que l'on minimise pour obtenir

$$x^{(k+1)} = x^{(k)} - [H(J(x^{(k)}))]^{-1} \nabla J(x^{(k)})$$

* Algo du gradient conjugué

Etape 1 : $x^{(0)}$, on calcul $d_0 = \nabla J(x^{(0)})^T$

A l'itération k, on calcul la valeur du pas

$$\rightarrow h_k = \frac{d_{k-1}^T \cdot \nabla^T J(x^{(k-1)})}{d_{k-1}^T \cdot H(J(x^{(k-1)}))\, d_{k-1}}$$

on calcule le nouveau point

$$\rightarrow x^{(k)} = x^{(k-1)} - h_k d_{k-1}$$

on calcule la nouvelle direction

$$\rightarrow d_k = \nabla J(x^{(k)}) - \frac{H(J(x^{(k)}))}{H(J(x^{(k-1)}))} d_k$$

Arrêt $\| \nabla J(x^{(k)}) \| < \epsilon$ seuil.

* Algo du gradient conjugué appliqué à la résolution d'un système linéaire.

1) On choisit un vecteur initial $x^{(0)} \in \mathbb{R}^n$.
On calcule le résidu.

$$r_0 = A x^{(0)} - b.$$

et la direction

$$d_0 = r_0.$$

2) * A l'itération $k \geq 1$, on calcule la valeur du pas.

$$\rightarrow h_k = \frac{d_{k-1}^T \cdot r_{k-1}}{d_{k-1}^T A d_{k-1}}$$

* On calcule le nouveau point

$$\rightarrow x^{(k)} = x^{(k-1)} + h_k d_{k-1}$$

* On calcule le nouveau résidu

$$\rightarrow r_k = r_{k-1} - h_k A d_{k-1}$$

* On calcule la nouvelle direction

$$\rightarrow d_k = r_k - \frac{(A d_{k-1})^T r_k}{(A d_{k-1})^T d_{k-1}}$$

3) Arrêt lorsque

$$\| r_k \| < \varepsilon_{seuil}.$$

V) Deux pbs récurrents peuvent se poser lorsque on essaie de résoudre un système de grande taille :

1) la matrice n'est pas symétrique, ni définie positive

2) la matrice a un mauvais conditionnement

* Matrice non symétrique. $Ax = b$

↳ on résoud le système suivant :

$$A^T A x = A^T b \quad (\Leftrightarrow \text{moindres carrés})$$

↳ autre solution : on pose $x = A^T u$

$$\Rightarrow \quad A A^T u = b$$

⚠ Dans les deux cas, si $K(A)$ est grand (mauvais condit^{ot}) alors AA^T et A^TA sont pires !!!

$$K_2(A^TA) = K_2(A)^2$$

* Préconditionnement : On cherche C tq $\begin{cases} CA \text{ est bien conditionnée} \\ \text{produit } CA \text{ peut coûteux} \end{cases}$

• Décomposition : $C = (D + \omega L) D^{-1} (D + \omega U)$ avec $\omega \sim 1$

et $A = \begin{bmatrix} \ddots & & U \\ & D & \\ L & & \ddots \end{bmatrix}$

• Factorisation incomplète.

$$A = LU + R$$

$$\rightarrow \quad C = U^{-1} L^{-1}$$

• Inverse approchée : On cherche C qui minimise $\|I - CA\|$

$\infty \; \|I - AC\|$

* Gradient Conjugué Préconditionné

Préconditionnement : $Ax = b \Leftrightarrow \tilde{A}\tilde{x} = \tilde{b}$

où C est sym def positive :

$\tilde{A} = C^{-1} A C^{-1}$

$\tilde{x} = Cx$

$\tilde{b} = C^{-1} b$

Algorithme amélioré

On prend $M = C^2$ comme préconditionneur

et on note $z^{(k)}$ solution de $M z^{(k)} = r^{(k)}$

$x^{(0)}$ donné $\in \mathbb{R}^n$

$k = 0$.

Calculer $r^{(0)} = b - A x^{(0)}$

Tant Que $r^{(k)} \neq 0$ Faire

Résoudre $M z^{(k)} = r^{(k)}$

$k = k+1$

Si $k \neq 1$ alors

$$\beta_k = \frac{(r^{(k-1)})^T z^{(k-1)}}{(r^{(k-2)})^T z^{(k-2)}}$$

$$p^{(k)} = z^{(k-1)} + \beta_k p^{(k-1)}$$

Sinon

$$p^{(1)} = z^{(0)}$$

FinSi

$$\alpha_k = \frac{(r^{(k-1)})^T z^{(k-1)}}{(p^{(k)})^T A p^{(k)}}$$

$$x^{(k)} = x^{(k-1)} + \alpha_k p^{(k)}$$

$$r^{(k)} = r^{(k-1)} - \alpha_k A p^{(k)}$$

FinTantQue

$x = x^{(k)}$

* Raffinement itératif

↳ post-traitement des résultats pour corriger une partie des erreurs.

On suppose qu'au lieu de factoriser $A = LU$, on a factorisé
$$(A+E) = L_c U_c$$
et on a résolu exactement
$$(A+E)\, x_c = b.$$
$$\Rightarrow \quad r_c = b - A x_c \neq 0.$$

On pose $x^{(0)} = x_c$ et $r^{(0)} = r_c$. On introduit alors le processus suivant :
$$(A+E)\, e^{(k+1)} = r^{(k)}$$
$$x^{(k+1)} = \left(x^{(k)} + e^{(k+1)} \right)$$
$$r^{(k+1)} = \left(b - A x^{(k+1)} \right)$$

$$\Rightarrow \quad x - x^{(k+1)} = \left(I - (A+E)^{-1} A \right) \left(x - x^{(k)} \right)$$

Deux itérations de ce processus suffisent à nettement améliorer le résultat.

* Préconditionnement et erreurs de calcul.

- $$\frac{1}{K(A)} \cdot \frac{\| r^{(k)} \|}{\| r^{(0)} \|} \leq \frac{\| e^{(k)} \|}{\| e^{(0)} \|} \leq K(A) \frac{\| r^{(k)} \|}{\| r^{(0)} \|} \leq K^2(A) \frac{\| e^{(k)} \|}{\| e^{(0)} \|}$$

- Critère système d'arrêt : soit $x + \Delta x$ solut° du système perturbé $A + \Delta A$.
On calcule alors le résidu selon la formule suivante :
$$r = b - A(x + \Delta x)$$

On a la relation
$$\underbrace{\frac{\| r \|}{\| A \| \cdot \| x + \Delta x \|}}_{\text{↳ petit} \Rightarrow \text{résultat acceptable.}} \leq \frac{\| A + \Delta A \|}{\| A \|}$$