Soit $(\mathbb{K},+,\times)$ un corps commutatif

# I. Systèmes d'équations linéaires

## 1. Définitions et exemples [G]

**Déf** On appelle système d'équations linéaires un système du type :

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ \vdots \qquad\qquad \vdots \\ a_{p1}x_1 + \dots + a_{pn}x_n = b_p \end{cases}, \text{ d'inconnue } x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{K}^n,$$

où $(b_i)_{1\le i\le p} \in \mathbb{K}^p$ et $(a_{ij})_{\substack{1\le i\le p\\ 1\le j\le n}} \in \mathbb{K}^{np}$ sont fixés.

Un tel système est dit compatible s'il a une solution.

En notant $A = (a_{ij})$ et $b = (b_i)$, ce système se réécrit sous la forme matricielle $Ax = b$

**Déf** On appelle rang du système linéaire $Ax=b$ le rang de $A$, i.e la dimension de l'espace engendré par ses colonnes

**Ex** $\begin{cases} x+y=2 \\ y=1 \end{cases}$ est un système compatible de rang 2.

$\begin{cases} x+y=2 \\ x+y=0 \end{cases}$ n'est pas compatible.

**Rque** Résoudre un système linéaire à $p$ équations et $n$ inconnues revient à chercher l'intersection de $p$ hyperplans affines de $\mathbb{K}^n$

**Prop L** Le système $Ax=b$ est compatible ssi $b$ est engendré linéairement par les colonnes de $A$.

## 2. Systèmes de Cramer [O]

**Déf** Un système linéaire $Ax=b$ à $n$ équations et $n$ inconnues est dit de Cramer si $A$ est inversible. Un tel système admet toujours une unique solution $A^{-1}b$

**Th** (Cramer) En notant $C_1, \dots, C_n$ les colonnes de la matrice $A$, l'unique solution du système de Cramer $Ax=b$ est $(x_j)_{1\le j\le n}$, avec :

$$\forall j \in [\![1,n]\!],\ x_j = \frac{\det(C_1,\dots,C_{j-1},b,C_{j+1},\dots,C_n)}{\det A}$$

**Complexité** Déterminer la solution ainsi requiert le calcul de $n+1$ déterminants de taille $n$, ce qui s'effectue en $O(n^4)$

En pratique, on n'utilise pas cette formule, car on dispose de méthodes de résolution bien plus efficaces.

## 3. Conditionnement [All], [Cia]

Dans ce paragraphe, $\mathbb{K}=\mathbb{R}$ et $n=p$

**Déf** On munit $\mathcal{M}_n(\mathbb{R})$ d'une norme $\|\cdot\|$ subordonnée. On appelle conditionnement d'une matrice $A$ inversible le réel $\mathrm{cond}(A) = \|A\|\,\|A^{-1}\|$

**Prop** Soit $A \in GL_n(\mathbb{R})$. Soit $b \in \mathbb{R}^n$ non nul

- $\mathrm{cond}(A) = \mathrm{cond}(A^{-1}) \geq 1$
- Soient $x$ et $x+\delta x$ les solutions respectives de $Ax=b$ et $A(x+\delta x) = b+\delta b$. Alors : $\frac{\|\delta x\|}{\|x\|} \le \mathrm{cond}(A)\frac{\|\delta b\|}{\|b\|}$
- Soient $x$ et $x+\delta x$ les solutions respectives de $Ax=b$ et $(A+\delta A)(x+\delta x) = b$

$$\text{Alors } \frac{\|\delta x\|}{\|x+\delta x\|} \leq \mathrm{Cond}(A) \frac{\|\delta A\|}{\|A\|}$$

Ainsi, le conditionnement permet de contrôler l'erreur commise sur la solution en fonction de celle commise sur les données lors de calculs approchés effectués par ordinateur. Il est donc préférable d'avoir un conditionnement proche de 1.

## II. Opérations élémentaires

**Déf** Les opérations élémentaires sur les lignes et colonnes d'une matrice sont :

- Intervertir deux lignes (ou colonnes), ce qui revient à multiplier à gauche (resp. droite) par une matrice de transposition (On note $L_i \leftrightarrow L_j$ ou $C_i \leftrightarrow C_j$)
- Multiplier une ligne (ou colonne) par un scalaire non nul, ce qui revient à multiplier à gauche (resp. droite) par une matrice de dilatation (On note $L_i \leftarrow \alpha L_i$ ou $C_i \leftarrow \alpha C_i$ $\alpha \in \mathbb{K}^*$)
- Ajouter à une ligne (ou colonne) une combinaison linéaire des autres lignes (resp. colonnes), ce qui revient à multiplier à gauche (resp. droite) par une matrice de transvection (On note $L_i \leftarrow L_i + \alpha_j L_j$ ou $C_i \leftarrow C_i + \alpha_j C_j$

**Prop** Les opérations élémentaires sur les lignes d'un système linéaire n'en modifient pas les solutions

### 1. L'algorithme du pivot de Gauss [H2]

**Déf** On appelle pivot d'une ligne non-nulle le coefficient non-nul le plus à gauche

**Algo** (pivot de Gauss)

- Par une éventuelle interversion de ligne et de colonne, amener un coefficient non nul a en position (1, 1)
- $L_1 \leftarrow \frac{1}{a} L_1$
- Pour $i \in [\![2, p]\!]$, $L_i \leftarrow L_i - a_{i1} L_1$

  Pour $j \in [\![2, n]\!]$, $C_j \leftarrow C_j - a_{1j} C_1$ $\quad \begin{pmatrix} 1 & 0 \cdots 0 \\ 0 & \\ \vdots & A' \\ 0 & \end{pmatrix}$

Appliquer les étapes précédentes à la matrice $A'$ de taille $(p-1, n-1)$

**Prop** La complexité de cet algorithme est $O(n^3)$

**Rque** Des problèmes de conditionnement peuvent survenir si les pivots par lesquels on divise deviennent très petits.

**Applications**
- Le groupe $SL_n(\mathbb{R})$ est engendré par les transvections
- Calculs de déterminants
- Calcul de l'inverse d'une matrice

### 2. Action de $GL_p(\mathbb{K})$ à gauche [H2]

**Déf** Une matrice de $M_{p,n}(\mathbb{K})$ est dite échelonnée si :
- Toutes les lignes en dessous d'une ligne nulle sont nulles
- Le pivot d'une ligne est plus à droite strictement que les pivots des lignes précédentes.

Si de plus les pivots sont égaux à 1 et sont les seuls coefficients non nuls de leur colonne, on parle de matrice réduite

Ex $\begin{pmatrix} 0 & 1 & 0 & 2 & 0 & 3 & 4 \\ 0 & 0 & 1 & 4 & 0 & 3 & 3 \\ 0 & 0 & 0 & 0 & 1 & 2 & 2 \end{pmatrix}$ est réduite

**Th** On considère l'action de $GL_p(\mathbb{R})$ à gauche sur $\mathcal{M}_{p,n}(\mathbb{R})$. Alors toute matrice est dans l'orbite d'une matrice échelonnée réduite.

On dit que les matrices échelonnées réduites sont les formes normales de cette action

## 3. Action à gauche et à droite [H2]

**Th** (du rang) Deux matrices de $\mathcal{M}_{p,n}(\mathbb{K})$ sont équivalentes ssi elles ont même rang

**Cor** Deux matrices sont dans la même orbite sous l'action de $GL_p(\mathbb{K}) \times GL_n(\mathbb{K})$ ssi elles ont même rang

**Applications**
- Calcul pratique du rang d'une matrice
- Recherche d'une base d'un s.e.v défini par un système linéaire

## 4. Similitude et forme de Hessenberg [H2]

On considère l'action $GL_n(\mathbb{K}) \times \mathcal{M}_n(\mathbb{K}) \to \mathcal{M}_n(\mathbb{K})$ dont les orbites sont les classes de similitude.

$$(P, A) \mapsto PAP^{-1}$$

**Prop** Toute matrice $A$ de $\mathcal{M}_n(\mathbb{K})$ est semblable à une matrice de la forme $\begin{pmatrix} h_{11} & \cdots & h_{1n} \\ h_{21} & & \vdots \\ 0 & \ddots & \\ 0 & h_n & h_{nn} \end{pmatrix}$ dite forme de Hessenberg de $A$

La forme de Hessenberg donne notamment une méthode efficace pour calculer le polynôme caractéristique (récursivement, en développant par rapport à la dernière colonne)

## 5. Décompositions LU et de Cholesky

**Th** Soit $A \in GL_n(\mathbb{R})$ tel que $\forall k \in [\![1, n]\!]$, $\begin{vmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{vmatrix} \neq 0$

Alors il existe $L$ tinf et $U$ tsup tels que : $A = LU$

**Rque** Si la condition sur les mineurs n'est pas remplie, on peut s'y ramener par une série d'opérations élémentaires idoines.

**Rque** Cette décomposition permet de ramener la résolution de $Ax = b$ à la résolution de deux systèmes triangulaires.

**Th** (Cholesky) Toute matrice $A$ définie positive se factorise sous la forme $A = B^tB$ avec $B$ tsup. Cette décomposition est unique si on impose que la diagonale de $B$ soit positive

**Rque** Cette décomposition permet de ramener la résolution d'un système $Ax = b$ avec $A$ symétrique définie positive, à la résolution successive de deux systèmes triangulaires

## 6. Facteurs invariants

**Prop** Soit $(A, +, \times, S)$ un anneau euclidien. Soit $U \in \mathcal{M}_{m,n}(A)$. Alors il existe $d_1 | d_2 | \cdots | d_s \in (A^*)^s$ tel que $A$ soit semblable à $\begin{pmatrix} d_1 & & 0 \\ & d_s & \\ 0 & & 0 \end{pmatrix}$

## III. Méthodes de résolution itératives [Cia, H2]

### 1. Méthodes de Jacobi, de Gauss-Seidel et de relaxation

Idée générale Pour résoudre le système $Ax = b$, on décompose $A = M - N$, avec $M$ facile à inverser.

On calcule alors la suite des itérés $x_{k+1} = M^{-1}Nx_k + M^{-1}b$

La limite, si elle existe, ne peut être alors que la solution $x$ du système.

Décomposons $A$ en $A = D - E - F$ où $D$ est la diagonale de $A$, $-E$ sa partie t sup et $-F$ sa partie t inf.

| Méthode | Décomposition M-N | $M^{-1}N$ | Description d'une itération |
|---|---|---|---|
| Jacobi | $A = D - (E+F)$ | $D^{-1}(E+F)$ | $Dx_{k+1} = (E+F)x_k + b$ |
| G-S | $A = (D-E) - F$ | $(D-E)^{-1}F$ | $(D-E)x_{k+1} = Fx_k + b$ |
| Relaxation $\omega \neq 0$ | $A = \left(\frac{1}{\omega}D - E\right) - \left(\frac{1-\omega}{\omega}D + F\right)$ | $\left(\frac{D}{\omega} - E\right)^{-1}\left(\frac{1-\omega}{\omega}D + F\right)$ | $\left(\frac{D}{\omega} - E\right)x_{k+1} = \left(\frac{1-\omega}{\omega}D + F\right)x_k + b$ |

Dans tous les cas, on part d'un point $x_0$ arbitraire de l'espace

Prop Si $A = M - N$, avec $M$ inversible, est symétrique définie positive ainsi que ${}^tM + N$, alors $\rho(M^{-1}N) < 1$ et donc la suite des itérés
$$\begin{cases} x_0 \\ x_{k+1} = M^{-1}Nx_k + M^{-1}b \end{cases}$$
converge

Références [G] Grifone, Algèbre linéaire [All] Allaire [D] Demailly
[Cia] Ciarlet [H2] Histoires hédonistes

### 2. Méthodes de gradient

Soit $A$ une matrice réelle symétrique définie positive

On s'intéresse à l'application
$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$
$$x \longmapsto \frac{1}{2}(Ax|x) - (x|b)$$
, qui est strictement convexe, dérivable et de gradient $\nabla f(x) = Ax - b$.

Son unique minimum est donc la solution de $Ax = b$.

Algo (gradient à pas optimal)

$x_0 \in \mathbb{R}^n$

$$\forall k \begin{cases} t_k \text{ choisi minimisant } f(x_k - t\nabla f(x_k)) \\ x_{k+1} = x_k - t_k \nabla f(x_k) \end{cases}$$

Def Deux vecteurs $x$ et $y$ sont dits conjugués si $(x|Ay) = 0$

Algo (gradient conjugué)

- $x_0 \in \mathbb{R}^n$, $d_0 = \nabla f_{x_0}$, $\alpha_0 = \frac{(\nabla f_{x_0}|d_0)}{(Ad_0|d_0)}$, $x_1 = x_0 - \alpha_0 d_0$
  $k = 1$
- Tant que $\nabla f_{x_k} \neq 0$

$$d_k \leftarrow \nabla f_{x_k} + \frac{\|\nabla f_{x_k}\|^2}{\|\nabla f_{x_{k-1}}\|^2} d_{k-1}$$

$$\alpha_k \leftarrow \frac{(\nabla f_{x_k}|d_k)}{(Ad_k|d_k)}$$

$$x_{k+1} \leftarrow x_k - \alpha_k d_k \ ; \ \text{incr } k$$

DVT

Th Cet algorithme converge en au plus $n$ itérations vers $A^{-1}b$

De fait, $\Im(h(z))$ est du signe de $-bcy + ady = (ad - bc)\Im(y) > 0$ car $ad - bc > 0$. Ainsi, par stabilité, $PSL_2(\mathbb{R})$ agit sur $\mathcal{H}$. Enfin, $PSL_2(\mathbb{R})$ contient les translations et homothéties de rapport positif, d'où la transitivité (faire un dessin, sans oublier que l'on ne part jamais vraiment de l'axe réel donc que les homothéties positives font remonter "en diagonale").

(*ii*) Comme les homographies conservent les cercles ou droites, $PGL_2(\mathbb{R})$ agit naturellement sur l'ensemble des cercles ou droites de $\mathbb{P}^1(\mathbb{C})$. De plus, les homographies positives conservent $\mathcal{H}$ (cf. supra), donc on a bien une action de $PSL_2(\mathbb{R})$ sur $D_h$. Soit $z \in \mathcal{H}$ et soit $D \in D_h$ passant par $z$. D'après le point $(i)$, il existe $g \in PSL_2(\mathbb{R})$ tel que $g(z) = i$ : $g(D)$ est alors une droite hyperbolique passant par $i$.

Soit $D_i$ une droite hyperbolique passant par $i$ telle que $D_i \neq i\mathbb{R}_+^*$ : c'est alors un demi-cercle coupant l'axe réel en deux points; considérons l'un de ces points, soit $t$. Il existe alors un unique réel $\theta \in (-\pi, \pi)$ tel que $\tan(\theta) = t$. On note alors $h_\theta \in PSL_2(\mathbb{R})$ l'homographie associée à la rotation $\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \in SL_2(\mathbb{R})$ et on remarque que :

$$\begin{aligned} h_\theta(i) &= \frac{i\cos(\theta) - \sin(\theta)}{i\sin(\theta) + \cos(\theta)} \\ &= e^{-i\theta}(i\cos(\theta) - \sin(\theta)) \\ &= e^{-i\theta}i(\cos(\theta) + \sin(\theta)) \\ &= i \end{aligned}$$

Ainsi $h_\theta \in \mathrm{Stab}_{PSL_2(\mathbb{R})}(i)$. De plus :

$$\begin{aligned} h_\theta(t) &= \frac{t\cos(\theta) - \sin(\theta)}{t\sin(\theta) + \cos(\theta)} \\ &= \frac{\tan(\theta)\cos(\theta) - \sin(\theta)}{t\sin(\theta) + \cos(\theta)} \\ &= 0 \end{aligned}$$

In fine, si on note $D_i'$ le cercle ou droite correspondant à $D_i$ dans $\mathbb{P}^1(\mathbb{C})$, $h_\theta(D_i')$ passe par $i$ et 0. Comme $h_\theta(D_i)$ est une droite hyperbolique, on a nécessairement $h_\theta(D_i') = i\mathbb{R}$ et donc $h_\theta(D_i) = i\mathbb{R}_+^*$.

Ainsi, quitte à remplacer $g$ par une composée de la forme $h_\theta \circ g$, avec $\theta$ bien choisi, on a $g(D) = i\mathbb{R}_+^*$ : l'action considérée est bien transitive.

**Détails supplémentaires :**

- Soient quatre points $a, b, c, d \in \mathbb{P}^1(\mathbb{C})$ ; on suppose $[a, b, c, d] \in \mathbb{R}$. Notons $h$ l'homographie définissant ce birapport, i.e telle que $h(a) = 0, h(b) = 1, h(c) = \infty$. Alors $h(a), h(b), h(c)$ et $h(d) = [a, b, c, d]$ sont alignés et comme l'homographie $h^{-1}$ conserve les cercles ou droites on a bien que $a, b, c$ et $d$ sont alignés ou cocycliques.

## 1.2 Algorithme des facteurs invariants

Référence :
- [BMP05], p. 285–288

On se propose de démontrer de façon algorithmique l'existence[1] de la décomposition suivante :

**Proposition 1.2**
*Soit* $(\mathbb{A}, \delta)$ *un anneau euclidien.*
*Soit* $U \in \mathcal{M}_{m,n}(\mathbb{A})$.

1. Le lecteur averti aura remarqué qu'on peut également énoncer un résultat d'unicité de cette dernière. On peut d'ailleurs en trouver une démonstration dans [BMP05], p. 289.

*Alors il existe* $d_1, \dots, d_s \in \mathbb{A}^*$ *tels que* $\forall i \in [s-1]$, $d_i | d_{i+1}$ *et* $(P, Q) \in GL_m(\mathbb{A}) \times GL_n(\mathbb{A})$ *tels que :*

$$U = PDQ, \text{ où } D := \begin{pmatrix} d_1 & & & (0) \\ & \ddots & & \\ & & d_s & \\ & & & (0) \\ (0) & & & \end{pmatrix}$$

On considère donc l'algorithme[2] suivant :

**Algorithme 1.1 (Facteurs invariants)**
Entrée *:* $U \in \mathcal{M}_{m,n}(\mathbb{A})$
Sortie *:* $D \in \mathcal{M}_{m,n}(\mathbb{A})$ *correspondant à la réduction de U donnée dans la proposition 1.2.*
*On suit alors les étapes suivantes :*

1. Si $U = 0$, *renvoyer* $U$. Sinon, *passer à l'étape* 2.
2. Choix d'un pivot. *Sélectionner* $(i_0, j_0) \in [m] \times [n]$ *tel que* $\delta(u_{i_0,j_0}) = inf\{\delta(u_{i,j}) \,|\, u_{i,j} \neq 0\}$. *Effectuer ensuite les opérations élémentaires suivantes :*
$$C_1 \leftrightarrow C_{j_0} \tag{1.2.1}$$
$$L_1 \leftrightarrow L_{i_0} \tag{1.2.2}$$
3. Traitement de la première colonne. *Initialiser un compteur* $i \leftarrow 2$.
   (a) *Par division euclidienne, on peut écrire (dans* $\mathbb{A}$*) :*
   $$u_{i,1} = u_{1,1}q + r_i, \text{ avec } r_i = 0 \text{ ou } \delta(r_i) < \delta(u_{1,1})$$
   *Effectuer alors l'opération suivante :*
   $$L_i \leftarrow L_i - qL_1 \tag{1.2.3}$$
   *On a ainsi remplacé* $u_{i,1}$ *par* $r_i$.
   (b) Si $r_i \neq 0$, *effectuer l'opération suivante :*
   $$L_i \leftrightarrow L_1 \tag{1.2.4}$$
   *Retourner ensuite en* 3.(a).
   (c) Si $r_i = 0$ *et* $i < m$, *incrémenter* $i$ $(i \leftarrow i + 1)$ *et retourner en* 3.(a).
   (d) Sinon *passer à l'étape* 4. *On a désormais :*
   $$C_1 = \begin{pmatrix} u_{1,1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$
4. Traitement de la première ligne. *Initialiser un compteur* $j \leftarrow 2$.
   (a) *Par division euclidienne on a :*
   $$u_{1,j} = u_{1,1}q + s_j, \text{ avec } s_j = 0 \text{ ou } \delta(s_j) < \delta(u_{1,1})$$
   *Effectuer ensuite l'opération suivante :*
   $$C_j \leftarrow C_j - qC_1 \tag{1.2.5}$$
   *On a ainsi remplacé* $u_{1,j}$ *par* $s_j$.
   (b) Si $s_j \neq 0$, *effectuer l'opération suivante :*
   $$C_j \leftrightarrow C_1 \tag{1.2.6}$$
   *Retourner ensuite en* 4.(a).

2. On ne se privera donc pas de faire usage des conventions d'écriture propres à l'algorithmique, par exemple en continuant de noter $U$ notre matrice quelques soient les sévices que nous lui ferons subir.

(c) Si $s_j = 0$ *et* $j < n$, *incrémenter* $j$ *et retourner en* 4.(a).

(d) Sinon *passer à l'étape* 5. *On a désormais :*

$$U = \begin{pmatrix} u_{1,1} & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & & \\ \vdots & & (*) & \\ 0 & & & \end{pmatrix}$$

5. Vérification de la divisibilité.

(a) Si *il existe* $i_1, j_1 \geq 2$ *tels que* $u_{1,1} \nmid u_{i_1,j_1}$, *effectuer l'opération suivante :*

$$C_1 \leftarrow C_1 + C_{j_1} \tag{1.2.7}$$

*Retourner ensuite en* 3.

(b) Sinon *retourner en* 1 *avec* $U \leftarrow (u_{i,j})_{2\leq i\leq m, 2\leq j\leq n}$ *(i.e on finit le traitement de façon récursive).*

TERMINAISON : L'idée est ici de montrer que chaque retour en arrière fait strictement décroître l'entier $\delta(u_{1,1})$. On conclura alors qu'il n'est possible d'effectuer qu'un nombre fini d'étapes et donc que l'algorithme termine.

Les étapes 1 et 2 ne posent pas de problèmes[3]. Lors de l'étape 3, on ne rebrousse chemin que si $r_i \neq 0$ (étape 3.(b)) et alors on remplace $u_{1,1}$ par $r_i$ ce que implique une décroissance stricte de $\delta(u_{1,1})$. Ceci ne pouvant de facto arriver qu'un nombre fini de fois, on passera[4] nécessairement à l'étape 4.

Durant l'étape 4, on ne retournera en arrière qu'à l'expresse condition que $s_j \neq 0$ (4.(b)). Comme précédemment, on en déduit qu'on passera nécessairement à l'étape 5.

Passer de l'étape 5 à une étape antérieure (en l'occurence 3) requière qu'il $i_1, j_1 \geq 2$ tels que $u_{1,1} \nmid u_{i_1,j_1}$. On remplace alors $C_1$ par $C_1 + C_{j_1}$ ce qui, au vu de l'état de notre matrice $U$, laisse $u_{1,1}$ inchangé et remplace $u_{i_1,1}$ par $u_{i_1,j_1}$ et donc $r_{i_1} \neq 0$. Ainsi ce passage par l'étape 3 verra strictement décroître $\delta(u_{1,1})$ ; on ne pourra donc se permettre qu'un nombre fini de telles excentricités, entraînant que l'algorithme termine.

CORRECTION : Au bout d'une itération de l'algorithme (i.e avant de poursuivre récursivement) on obtient une matrice $U'$ de la forme suivante[5] :

$$U' = \begin{pmatrix} u'_{1,1} & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & & \\ \vdots & & U_1 & \\ 0 & & & \end{pmatrix}$$

L'étape 5 nous assure par ailleurs que $u'_{1,1}$ divise tous les coefficients de $U_1$. Chaque étape de notre algorithme n'étant au final, une fois les tests conditionnels et autres fioritures éliminées, qu'une succession d'opérations élémentaires sur les lignes et colonnes, il existe $(P, Q) \in GL_m(\mathbb{A}) \times GL_n(\mathbb{A})$ telles que $U = PU'Q$.

De même, il va exister $(P_1, Q_1) \in GL_{m-1}(\mathbb{A}) \times GL_{n-1}(\mathbb{A})$ telles que $U_1 = P_1 U'_1 Q_1$, où :

$$U'_1 = \begin{pmatrix} u''_{2,2} & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & & \\ \vdots & & U_2 & \\ 0 & & & \end{pmatrix}$$

3. Ou alors c'est que vous vous y prenez *très* mal ...
4. Un jour ...
5. Ce paragraphe contenant un raisonnement mathématique, il n'est plus réellement tolérable de continuer à la noter $U$. Ceci étant, chacun reste libre d'en faire à sa guise.

Ainsi, si on pose :

$$P_1' = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & P_1 & \\ 0 & & & \end{pmatrix} \quad \text{et} \quad Q_1' = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & Q_1 & \\ 0 & & & \end{pmatrix}$$

On a :

$$U = PP_1' \begin{pmatrix} u'_{1,1} & 0 & \dots & 0 \\ 0 & u''_{2,2} & & \\ \vdots & & & \\ \vdots & & U_2 & \\ 0 & & & \end{pmatrix} Q_1'Q$$

Comme nous n'avons effectué que des opérations élémentaires que que $u_{1,1}$ divisait tous les coefficients de $U_1$ il divise tous ceux de $U_2$ ainsi que $u''_{2,2}$. De plus l'étape 5 (itération 2) nous assure que $u''_{2,2}$ divise tous les coefficients de $U_2$. Par récurrence (utiliser l'invariant de boucle que l'on vient d'énoncer concernant la divisibilité), l'algorithme est correct.

## 1.3 Décomposition de Dunford

Références :
- [FGN09a] p.134–135 et [BMP05] p. 210–211 et 215–216.

**Proposition 1.3 (Dunford)**
*Soit $\mathbb{K}$ un corps.*
*Soit $E$ un $\mathbb{K}$-e.v de dimension finie.*
*Soit $f \in \mathcal{L}(E)$.*
*On suppose que $\chi_f$ est scindé sur $\mathbb{K}$.*
*Alors il existe un unique couple $(d, n) \in \mathcal{L}(E)^2$ tel que :*

*(i) $d$ soit diagonalisable ;*

*(ii) $n$ soit nilpotent ;*

*(iii) $f = d + n$ ;*

*(iv) $n \circ d = d \circ n$.*

*De plus, $d$ et $n$ sont alors des polynômes en $f$.*

DÉMONSTRATION :
- *Existence.* Comme $\chi_f$ est scindé on peut l'écrire sous la forme :

$$\chi_f = (-1)^n \prod_{i=1}^{s} (X - \lambda_i)_i^{\alpha}$$

Si on pose pour $i \in [s]$ $N_i := \ker(f - \lambda_i \mathrm{id}_E)^{\alpha_i}$ on a alors par lemme des noyaux appliqué à $\chi_f$ :

$$E = \ker(\chi_f(f)) = \bigoplus_{i=1}^{s} N_i$$

On définit alors $d$ et $n$ de la façon suivante :

$$\forall i \in [s], \forall x \in N_i, \begin{cases} d(x) := \lambda_i x \\ n(x) := f(x) - \lambda_i x \end{cases}$$

Il est alors clair que $d + n = f$, que $d$ est diagonalisable (prendre une base adaptée à la décomposition $E = \oplus_{i=1}^{s} N_i$) et que :

$$\forall i \in [s], \forall x \in N_i, n^{\alpha_i}(x) = (f - \lambda_i \mathrm{id}_E)_i^{\alpha}(x) = 0$$

## 1.2 Algorithme du gradient conjugué

### 1.2.1 Résultats

On cherche à résoudre le problème inverse $Ax = b$ pour $x \in \mathbb{R}^n$, ou de façon équivalente à résoudre

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle \tag{1.6}$$

où $A$ est symétrique définie positive. Comme $f$ strictement convexe de gradient $\nabla f_x = Ax - b$, l'unique minimum de $f$ sera la solution de $Ax = b$.

L'algorithme du gradient conjugué s'écrit

- Initialisations : $k = 0$, $u_0 \in \mathbb{R}^n$, $d_0 = \nabla f_{u_0}$, $\alpha_0 = \frac{\langle \nabla f_{u_0}, d_0\rangle}{\langle Ad_0, d_0\rangle}$, et $u_1 = u_0 - \alpha_0 d_0$ ;
- Tant que $\nabla f_{u_k} \neq 0$,
  - $d_k = \nabla f_{u_k} + \frac{\|\nabla f_{u_k}\|^2}{\|\nabla f_{u_{k-1}}\|^2} d_{k-1}$
  - $\alpha_k = \frac{\langle \nabla f_{u_k}, d_k\rangle}{\langle Ad_k, d_k\rangle}$
  - $u_{k+1} = u_k - \alpha_k d_k$
- Retourner $u_k$

**Définition 1.2.1.** On dit que deux vecteurs $x$ et $y$ sont conjugués par rapport à $A$ si et seulement si $\langle x, Ay\rangle = 0$.

**Théorème 1.2.1.** *Si $A$ une matrice symétrique définie positive, l'algorithme du gradient conjugué converge en au plus $n$ itérations vers l'unique solution du problème (1.6).*

PREUVE : On construit un algorithme dans lequel on note $d_k$ les directions de descente successives :

$$u_{k+1} = u_k + \alpha_k d_k$$

**Etape 1 : Calcul de $\alpha_k$**
Supposons que l'on a construit $u_0$, ... , $u_k$. On cherche $\alpha_k$ minimisant $f(u_k + \alpha d_k)$. En annulant la dérivée par rapport à $\alpha$, on obtient $0 = \langle d_k, \nabla f_{(u_k+\alpha d_k)}\rangle = \langle d_k, A(u_k + \alpha d_k) - b\rangle$. On a donc

$$\alpha_k = -\frac{\langle \nabla f_{u_k}, d_k\rangle}{\langle Ad_k, d_k\rangle} \quad \text{et} \quad \langle d_k, \nabla f_{u_{k+1}}\rangle = 0$$

où le dénominateur est non nul, car la matrice $A$ est supposée définie positive.

**Etape 2 : Calcul des directions de descente $d_k$.**
On choisit les $d_k$ de la forme

$$d_k = \nabla f_{u_k} + \beta_k d_{k-1} \tag{1.7}$$

Si on impose aux directions successives d'être conjuguées, on obtient

$$\beta_k = -\frac{\langle \nabla f_{u_k}, Ad_{k-1}\rangle}{\langle d_{k-1}, Ad_{k-1}\rangle}$$

**Etape 3 : Propriétés d'orthogonalité**
Montrons par récurrence que, pour tout $k \geqslant 1$, pour tout $0 \leqslant i < j \leqslant k$, on a :

$$\langle d_i, \nabla f_{u_j} \rangle = 0 \tag{1.8}$$
$$\alpha_i \neq 0 \tag{1.9}$$
$$\langle \nabla f_{u_i}, \nabla f_{u_j} \rangle = 0 \tag{1.10}$$
$$\langle d_i, Ad_j \rangle = 0 \tag{1.11}$$

- Initialisation avec $k = 1$.
  On a $d_0 = \nabla f_{u_0}$ est non nul, sinon $u_0$ est solution. Comme on l'a vu à l'étape 1, on a $\langle d_0, \nabla f_{u_1} \rangle = \langle \nabla f_{u_0}, \nabla f_{u_1} \rangle = 0$.
  $\alpha_0 = \frac{\langle \nabla f_{u_0}, d_0 \rangle}{\langle Ad_0, d_0 \rangle}$. Son numérateur vaut $\|\nabla f_{u_0}\|^2$, il est non nul, sinon $u_0$ est solution.
  Enfin, on a choisi $\beta_0$ de telle sorte que $\langle d_1, Ad_0 \rangle = 0$.

- Montrons que les propriétés se propagent. On suppose que les relations sont vérifiées jusqu'à l'ordre $k$ inclus. On les étend pour l'indice $k+1$ en supposant que $\nabla f_{u_{k+1}} \neq 0$.

  On remarque que $\nabla f_{u_{k+1}} = Au_{k+1} - b = Au_k + \alpha_k Ad_k - b$, et donc

$$\nabla f_{u_{k+1}} = \nabla f_{u_k} + \alpha_k Ad_k \tag{1.12}$$

  · Montrons (1.8). Le choix de $\alpha_k$ impose $\langle d_k, \nabla f_{u_{k+1}} \rangle = 0$, comme on l'a vu à l'étape 1. Lorsque $i < k$, on a $\langle d_i, \nabla f_{u_{k+1}} \rangle = \langle d_i, \nabla f_{u_k} + \alpha_k Ad_k \rangle = 0$, par hypothèse de récurrence.
  · Montrons (1.9). $\alpha_{k+1} = -\frac{\langle \nabla f_{u_{k+1}}, d_{k+1} \rangle}{\langle Ad_{k+1}, d_{k+1} \rangle} = -\frac{\langle \nabla f_{u_{k+1}}, \nabla f_{u_{k+1}} + \beta_{k+1} d_k \rangle}{\langle Ad_{k+1}, d_{k+1} \rangle}$. Comme on a supposé que $\nabla f_{u_{k+1}} \neq 0$ et $A$ définie, on obtient

$$\alpha_{k+1} = -\frac{\|\nabla f_{u_{k+1}}\|^2}{\langle Ad_{k+1}, d_{k+1} \rangle} \neq 0$$

  · Montrons (1.10). Lorsque $i \leqslant k$, on a $\langle \nabla f_{u_i}, \nabla f_{u_{k+1}} \rangle = \langle d_i - \beta_i d_{i-1}, \nabla f_{u_{k+1}} \rangle = 0$, d'après (1.8).
  · Montrons (1.11). Lorsque $i = k$, on a $\langle d_{k+1}, Ad_k \rangle = 0$, par définition des $\beta_k$.
  Lorsque $i < k$, on obtient : $\langle d_{k+1}, Ad_i \rangle = \langle \nabla f_{u_{k+1}} + \beta_{k+1} d_k, Ad_i \rangle = \langle \nabla f_{u_{k+1}}, Ad_i \rangle$, par hypothèse de récurrence. Puis, $\langle \nabla f_{u_{k+1}}, Ad_i \rangle = \langle \nabla f_{u_{k+1}}, \frac{\nabla f_{u_{i+1}} - \nabla f_{u_i}}{\alpha_i} \rangle = 0$, d'après (1.10).

**Etape 4 : Expression des $\beta_k$**
En utilisant les relations (1.12) et (1.7) et les on a :

$$\begin{aligned}
\beta_k &= -\frac{\langle \nabla f_{u_k}, Ad_{k-1}\rangle}{\langle d_{k-1}, Ad_{k-1}\rangle} \\
&= -\frac{\langle \nabla f_{u_k}, \nabla f_{u_k} - \nabla f_{u_{k-1}}\rangle}{\langle d_{k-1}, \nabla f_{u_k} - \nabla f_{u_{k-1}}\rangle} \\
&= -\frac{\|\nabla f_{u_k}\|^2}{\langle d_{k-1}, \nabla f_{u_{k-1}}\rangle} \\
&= -\frac{\|\nabla f_{u_k}\|^2}{\langle \nabla f_{u_{k-1}} + \beta_{k-1} d_{k-2}, \nabla f_{u_{k-1}}\rangle} \\
&= -\frac{\|\nabla f_{u_k}\|^2}{\|\nabla f_{u_{k-1}}\|^2}
\end{aligned}$$

**Etape 5 : Conclusion**

Les directions de descente sont toutes conjuguées 2 à 2. Elles forment donc une base de $\mathbb{R}^n$ (car la matrice $A$ est symétrique définie positive). On note $u^*$ la solution de notre problème.
D'une part, les coordonnées de $u^* - u_0$ dans la base des directions de descente. On a alors :

$$u^* - u_0 = \sigma_0 d_0 + \ldots + \sigma_{n-1} d_{n-1} \tag{1.13}$$

$$\sigma_k = \frac{\langle d_k, A(u^* - u_0)\rangle}{\langle d_k, Ad_k\rangle} \tag{1.14}$$

D'autre part, les itérations du gradient fournissent

$$u_k = u_0 + \alpha_0 d_0 + \ldots + \alpha_k d_k$$

En multipliant cette expression par $Ad_k$, on obtient $\langle d_k, A(u_k - u_0)\rangle = 0$, donc

$$\langle d_k, A(u^* - u_0)\rangle = \langle d_k, (Au^* - Au_k)\rangle = \langle d_k, (b - Au_k)\rangle = -\langle d_k, \nabla f_{u_k}\rangle$$

Finalement, on a bien $\alpha_k = \sigma_k$ pour tout $k$, ce qui achève la preuve du théorème. □

### 1.2.2 Commentaires

**Algorithme publié par Hestenes et Stiefel en 1952, "Methods of conjugate gradients for solving linear systems"**

Attention, ce ne sont pas les gradients qui sont conjugués, mais les directions de descente. Remarquons qu'une des bonnes propriétés de l'algorithme est son faible besoin de mémoire.

**Comparaison aux autres méthodes de descente :**
Dans l'algorithme de gradient à pas optimal, les gradients successifs sont orthogonaux. La méthode de relaxation consiste à choisir des directions de descente le long des axes de coordonnées. La méthode du gradient conjugué consiste à choisir des directions de descente conjugué par rapport à la matrice A. Ce qui a pour conséquence de rendre toutes les gradients en les itérés orthogonaux 2 à 2, et qui fournit la convergence en $n$ itérations de la méthode.