110

Caractères d'un groupe abélien fini et transformée de Fourier discrète. Applications.

Cadre : $G$ est un groupe abélien fini de cardinal $n \in \mathbb{N}^*$

# I) Caractères de groupe abélien et groupe dual.

## A) Définition et premier propriété

**Def 1 :** Un caractère $\chi$ de $G$ est un morphisme de $G$ dans $(\mathbb{C}^*, \times)$. On note $\hat{G}$ le dual de $G$ qui est l'ensemble des caractère de $G$.

**prop 2 :** Soit $\chi \in \hat{G}$, alors $\chi(G) \subset \mathbb{U}_n$.

**Rem 3 :** $\hat{G}$ est fini.

**Def 4 :** On note $\mathbb{C}[G]$ l'ensemble des fonctions de $G$ dans $\mathbb{C}$. C'est un espace vectoriel sur $\mathbb{C}$. On a produit scalaires hermitien défini par $\forall f, g \in \mathbb{C}[G]^2$,
$\langle f, g \rangle = \frac{1}{|G|} \sum_{x \in G} f(x)\, g(x)$.

**prop 5 :** $\delta_g(h) = \begin{cases} 1 \text{ si } h = g \\ 0 \text{ si } h \neq g \end{cases}$.
$(\delta_g)_{g \in G}$ est une base de $\mathbb{C}[G]$. En particulier $\dim_{\mathbb{C}} \mathbb{C}[G] = |G|$.

**prop 6 :** Soit $G$ un groupe cyclique de générateur $g$. On pose $\omega = e^{\frac{2i\pi}{n}}$. On a $\hat{G} = \{\chi_j, j \in [\![0; n-1]\!]\}$ où $\chi_j(g) = \omega^j$. En particulier $G \simeq \hat{G}$.

**prop 7 :** Si $G$ est cyclique alors $\hat{G}$ est une base orthonormale.

## B) Structure du dual.

**prop 8 :** Soit $H$ un sous groupe de $G$. Tout caractère $\chi$ de $H$ peut être prolongé en un caractère de $G$.

**Thm 9 :** $G$ et $\hat{G}$ ont le meme ordre.

**Def 10 :** On note $\hat{\hat{G}}$ le dual de $\hat{G}$, appelé le bidual de $G$.

**prop 11 :** $\hat{\hat{G}}$ est canoniquement isomorphe à $G$.

**prop 12 :** $G$ et $\hat{G}$ ont même exposant.

**Thm 13 :** (Structure des groupes abélien fini) Il existe $r \in \mathbb{N}$ et des entiers $n_1, \dots, n_r$ où $n_1$ est l'exposant de $G$, et $n_{i+1} | n_i$ si $i \leq r-1$ tel que $G \simeq \mathbb{Z}/n_1\mathbb{Z} \times \dots \times \mathbb{Z}/n_r\mathbb{Z}$.

**Thm 14 :** $G$ et son dual sont isomorphe

## C) Relation d'orthogonalité :

Lem 15: Soit $\chi \in \hat{G}$ on a
$$\sum_{g \in G} \chi(g) = \begin{cases} 0 \text{ si } \chi \neq 1 \\ |G| \text{ si } \chi = 1 \end{cases}$$

prop 16: $\hat{G}$ est une famille orthogonale.

Cor 17: $\hat{G}$ est une base orthogonal de $\mathbb{C}[G]$.

Prop 18: Soit $g, h \in G$. On a alors.
$$\sum_{\chi \in \hat{G}} \overline{\chi(g)}\, \chi(h) = \begin{cases} 0 \text{ si } g \neq h \\ |G| \text{ si } g = h \end{cases}$$
(*1) voir feuille suivante

## II) Transformée de Fourier discrète.

Def 19: Pour $f \in \mathbb{C}[G]$ on définit, pour $\chi \in \hat{G}$, le coefficient de Fourier $c_f(\chi) = \langle \chi, f \rangle$.
On a l'application $c : \begin{cases} \mathbb{C}[G] \longrightarrow \mathbb{C}[\hat{G}] \\ f \longmapsto c_f \end{cases}$.

Def 20 L'application transformée de Fourier définie par $\mathcal{F} : \begin{cases} \mathbb{C}[G] \longrightarrow \mathbb{C}[\hat{G}] \\ f \longmapsto \hat{f} \end{cases}$ ou
$\forall \chi \in \hat{G}$ $\hat{f}(\chi) = |G|\, c_f(\overline{\chi}) = \sum_{x \in G} f(x) \chi(x)$.

prop 21: (Formule d'inversion) Pour $f \in \mathbb{C}[G]$
$$f = \sum_{\chi \in \hat{G}} c_f(\chi)\, \chi = \frac{1}{|G|} \sum_{\chi \in \hat{G}} \hat{f}(\chi)\, \chi^{-1}$$

prop 22: (Isomorphisme de Fourier) c et $\mathcal{F}$ sont des isomorphisme de $\mathbb{C}[G]$ dans $\mathbb{C}[\hat{G}]$.

prop 23: (Formule de Plancherel) Pour $f, g \in \mathbb{C}[G]$ on a.
$$\sum_{x \in G} f(x) \overline{g(x)} = |G| \sum_{\chi \in \hat{G}} c_f(\chi)\, \overline{c_g(\chi)} = \frac{1}{|G|} \sum_{\chi \in \hat{G}} \hat{f}(\chi)\, \overline{\hat{g}(\chi)}$$

## B) Algèbre d'un groupe abélien.

Def 24: (Produit de convolution) Pour $f_1, f_2 \in \mathbb{C}[G]$ On pose $f_1 * f_2(g) = \sum_{h \in G} f_1(h) f_2(h^{-1} g)$

prop 25: $*$ est commutatif, associatif et bilinéaire. $(\mathbb{C}[G], +, *)$ est une algèbre

prop 26: Soit $f : G \to \mathbb{C}^*$ un morphisme de groupe. Il existe une unique façon de l'étendre en un morphisme d'algèbre $\tilde{f} : \mathbb{C}[G] \longrightarrow \mathbb{C}$.

prop 27: Soit $\chi \in \hat{G}$. $E_\chi : \begin{cases} \mathbb{C}[G] \longrightarrow \mathbb{C} \\ f \longmapsto \hat{f}(\chi) \end{cases}$ est l'unique façon d'étendre $\chi$ en un morphisme d'algèbre.

Thm 28: Pour $f, g \in \mathbb{C}[G]$ on a
$\widehat{f * g} = \hat{f} \cdot \hat{g}$ et $c_{f*g} = |G|\, c_f \cdot c_g$. En particulier $\mathcal{F}$ est un isomorphisme d'algèbre entre $(\mathbb{C}[G], +, *)$ et $(\mathbb{C}[\hat{G}], +, \cdot)$.

(*1) Ex : Soit $\varphi$ une fonction de $G^m$ dans $G$. Pour $h \in G$ on note $N(h)$ le nombres de $n$-uplets $(g_1 \cdots g_m)$ tels que $\varphi(g_1 \cdots g_m) = h$. On a :

$$N(h) = \frac{1}{|G|} \sum_{g_1 \in G} \cdots \sum_{g_m \in G} \sum_{\chi \in \hat{G}} \chi(\varphi(g_1 \cdots g_m)) \overline{\chi(h)}$$

Ex 29 : $f \in \mathbb{C}[G]$ non nulle on a.

$$|\mathrm{Supp}(f)| \times |\mathrm{Supp}(\hat{f})| \geq |G|.$$

où $|\mathrm{Supp}(f)|$ désigne la taille du support de $f$.

## III) FFT

On se place sur $G = \mathbb{Z}/N\mathbb{Z}$.

Def 30 : Soit $x \in \mathbb{C}^N$ on définit

$$\hat{x}_k = \sum_{n=0}^{N-1} x_n \omega_N^{nk} \quad \text{pour } k = 0, \ldots, N-1.$$

où $\omega_N = e^{\frac{2\pi i}{N}}$

Prop 31 : Pour $x \in \mathbb{C}^N$, $n \in [\![0; N-1]\!]$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}_k \omega_N^{-nk}.$$

Thm 32 : $F : x \longmapsto \hat{x}$ est un isomorphisme d'espace vectoriel de $\mathbb{C}^N$ dans $\mathbb{C}^N$.

FFT : prend $x \in \mathbb{C}^N$ renvoie $\hat{x} \in \mathbb{C}^N$ en $O(N \log_2 N)$.

Algorithme : pour $N = 2^m$ et $\omega = e^{\frac{2i\pi}{N}}$

fonction FFT($(a_0, \ldots a_{N-1}), \omega$)

Si $\omega = 1$ : renvoie $(a_0, \ldots, a_{N-1})$

$(s_0, s_1, \ldots, s_{\frac{n}{2}-1}) = \mathrm{FFT}((a_0, a_2, \ldots a_{n-2}), \omega^2)$

$(s'_0, s'_1, \ldots, s'_{\frac{n}{2}-1}) = \mathrm{FFT}((a_1, a_3, \ldots a_{n-1}), \omega^2)$.

Pour $j = 0$ à $\frac{n}{2} - 1$ :

$r_j = s_j + \omega^j s'_j$

$r_{j+\frac{n}{2}} = s_j - \omega^j s'_j$

renvoie $(r_0, \ldots, r_{n-1})$.

Appli 33 : multiplication de deux polynome en $O(N \log_2 N)$.

Référence :

G. Peyré, L'algèbre discrète de la transformée de Fourier

P. Colmez : Element d'analyse et d'algèbre.

S. Dasgupta, C Papadimitriou, U. Vazirani Algorithms

↳ remplacable par : T. Cormen, C Leiserson R. Rivest "Introduction à l'algorithmique.

# Un groupe abélien et son dual ont même ordre

Antoine LOUAZEL et Zoïs MOITIER

13 mai 2015

Référence : G. PEYRÉ, *L'algèbre discrète de la transformée de Fourier.* Ellipses. p.4-7.

**Propriétés.** *Soit $G$ un groupe abélien fini. Alors on a $|G| = |\widehat{G}|$.*

Pour démontrer cela on besion de deux résultats intermédiaires :
- Si on a un caractère d'un sous-groupe on peut le prolonguer en un caractère du groupe.
- Lemme technique.

**Propriétés** (Prolongement de caractères). *Soit $G$ un groupe abélien fini et $H$ un sous-groupe de $G$. Tout caractère $\chi$ de $H$ peut être prolongué en un caractère de $G$.*

*Démonstration.* On effectue une récurrence sur $[G : H]$.
Pour $[G : H] = 1$, on a $G = H$ donc la propriété est vraie.
On suppose que $[G : H] > 1$, il existe donc $x \in G$ tel que $x \notin H$. Soit $K = \langle x, H\rangle$ le groupe engendré par $x$ et $H$. Soit $m$ le plus petit entier non nul tel que $x^m \in H$, il existe car $x^n = 1 \in H$ où $n$ et le cardinal de $G$. Comme $G$ est abélien, tout éléments de $K$ s'écrit sous la forme $z = yx^k$ avec $y \in H$ et $k \in \{0, \dots, m-1\}$. De plus, cette écriture est unique. En effet, si $yx^k = y'x^{k'}$ avec $0 \leq k \leq k' \leq m - 1$ alors $x^{k-k'} = y'y^{-1} \in H$ or $k - k' < m$ donc $k = k'$ et $y = y'$.
Pour la suite, on procède par analyse sythèse :
**Analyse :** Supposon qu'il existe $\widetilde{\chi}$ un prolongement de $\chi$. Posons $\zeta = \widetilde{\chi}(x)$. Il nous faut $\zeta^n = \widetilde{\chi}(x^n) = 1$, d'où $\zeta$ est une racine $n$-ième de l'unité et $\zeta^m = \widetilde{\chi}(x^m) = \chi(x^m)$. On a alors si $z \in K$ s'écrit $z = yx^k$ avec $y \in H$ et $k \in \{0, ..., m-1\}$ :

$$\widetilde{\chi}(z) = \widetilde{\chi}(yx^k) = \chi(y)\zeta^k$$

**Syntèse :** Il existe $p \in \{0, \dots, |H| - 1\}$ tel que $\chi(x^m) = \mathrm{e}^{\frac{2i\pi p}{|H|}}$ d'où $\chi(x^m) = \mathrm{e}^{\frac{2i\pi p[G:H]}{n}}$ or $m|[G : H]$ car $m$ est l'ordre de $x$ dans $G/H$. On peut choisir $\zeta$ une racine $n$-ième de l'unité telle que $\chi(x^m) = \zeta^m$. Définissons, pour $z \in K$ décomposé sous la forme $z = yx^k$, le prolongement $\widetilde{\chi}$ par $\widetilde{\chi}(z) = \chi(y)\zeta^k$. On doit montrer que $\widetilde{\chi}$ est un élément de $\widehat{K}$. L'unicité de la décomposition montre que $\widetilde{\chi}$ est bien définie. Pour prouver le fait que c'est bien un morphisme, on prend $z = yx^k$ et $z' = y'x^{k'}$ deux éléments de $K$, et on distingue deux cas :
- Si $0 \leq k + k' \leq m - 1$, on a

$$\widetilde{\chi}(zz') = \widetilde{\chi}(yy'x^{k+k'}) = \chi(yy')\zeta^{k+k'} = \chi(y)\zeta^k\chi(y')\zeta^{k'} = \widetilde{\chi}(z)\widetilde{\chi}(z')$$

- Si $m \leq k + k' \leq 2m - 1$, on a

$$\widetilde{\chi}(zz') = \widetilde{\chi}(yy'x^m x^{k+k'-m}) = \chi(y)\chi(y')\chi(x^m)\zeta^{k+k'-m} = \widetilde{\chi}(z)\widetilde{\chi}(z')$$

On a $H$ sous-groupe de $K$ et $K$ sous-groupe de $G$ donc $[G : H] = [G : K][K : H]$ et $[K : H] > 1$. On a donc $[G : K] < [G : H]$. On peut alors avec l'hypothèse de récurrence, prolonger $\widetilde{\chi}$ à $G$. □

**Lemme.** *Soit $H$ un sous-groupe de $G$ un groupe abélien fini. On note $\rho : \widehat{G} \to \widehat{H}$ le morphisme de restriction et $j : \widehat{G/H} \to \widehat{G}$ le morphisme d'extension, défini par :*

$$\begin{array}{ccc} j : \widehat{G/H} & \longrightarrow & \widehat{G} \\ \chi & \longmapsto & \widetilde{\chi} \end{array} \quad \text{avec } \widetilde{\chi}(x) := \chi(xH)$$

*On a la suite exacte :*

$$\{1\} \to \widehat{G/H} \overset{j}{\hookrightarrow} \widehat{G} \overset{\rho}{\twoheadrightarrow} \widehat{H} \to \{1\}$$

*Ce qui donne $|\widehat{G}| = |\widehat{H}||\widehat{G/H}|$.*

*Démonstration.* $j$ est injective par définition, $\rho$ est surjective d'après la propriété de prolongement des caractères.
Si on considère $\chi \in \ker(\rho)$, alors $H \subset \ker(\chi)$, et donc par la propriété universelle du quotient, il existe $\widetilde{\chi} \in \widehat{G/H}$ tel que $\widetilde{\chi}(xH) = \chi(x)$, c'est à dire $j(\widetilde{\chi}) = \chi$, donc $\ker(\rho) \subset \mathrm{Im}(j)$. Réciproquement, un élément de $\mathrm{Im}(j)$ est trivial sur $H$. Donc on a bien $\ker(\rho) = \mathrm{Im}(j)$.
On définit :

$$\begin{array}{ccc} \widetilde{\rho} : \widehat{G}/j(\widehat{G/H}) & \longrightarrow & \widehat{H} \\ \chi\, j(\widehat{G/H}) & \longmapsto & \rho(\chi) \end{array}$$

$\widetilde{\rho}$ est bien défini par exactitude de la suite, elle est surjective car $\rho$ est surjective et est injective car $\ker(\rho) = j(\widehat{G/H})$, donc elle est bijective et on a $|\widehat{G}| = |\widehat{H}||\widehat{G/H}|$. □

On peut mainetant démontrer la propriété énoncé.

*Démonstration.* On raisonne par récurrence sur $n = |G|$.
Pour $n = 1$, le résultat est trivial car $\widehat{G} = \{1\}$, où 1 est le caractère triviale.
On suppose $n \geq 2$, il existe un groupe cyclique non triviale $H \subset G$. Si $H = G$, le résultat est vrai d'après la propriété sur le cas cyclique. Sinon avec l'hypothèse de récurrence, on a $|H| = |\widehat{H}|$ et $|G/H| = |\widehat{G/H}|$, et le lemme nous dit que $|\widehat{G}| = |\widehat{H}||\widehat{G/H}|$. Donc on a $|\widehat{G}| = |H||G/H| = |G|$. □

Commentaire : Attention pour la démonstration du prolongement des carctères faite dans G.Peyré, il dit que $\zeta^m = \chi(x^m) = \chi(1) = 1$ ce qui est faux.

# La FFT (Fast Fourier Transform) pour multiplier deux polynômes

Antoine LOUAZEL et Zoïs MOITIER

13 mai 2015

Référence : S. DASGUPTA, C. PAPADIMITRIOU, U. VAZIRANI, *Algorithms*. Broché. p.59-70.

Le but est de trouver un algorithme qui va multiplier deux polynômes.
- Naïvement on fait ça en $O(n^2)$ ou $n$ est le max des degré de $A$ et $B$
- la FFT le fait en $O(n\log(n))$

On a deux façons de représenter un polynôme de degré $n$ :
- par ces coefficients dans une base (par exemple $A = a_0 + \cdots + a_nX^n$)
  $\rightarrow$ multiplication en $O(n^2)$ ($c_n = \sum_{k=0}^{n} a_k b_{n-k}$)
- par ces valeurs en $d+1$ points distincts
  $\rightarrow$ multiplication en $O(n)$ ($C(x_k) = A(x_k)B(x_k)$)

On va donner deux polynômes $A = a_0 + \cdots + a_dX^d$ et $B = b_0 + \cdots + b_dX^d$. On peut les prendre de même degré sans perte de généralité quitte à ajouté un coefficient dominant nul. On cherche $C = c_0 + \cdots + c_dX^{2d} = AB$. Les polynômes serons représentés par les vecteurs $(a_0, \ldots, a_d)$, $(b_0, \ldots, b_d)$ et $(c_0, \ldots, c_{2d})$.
On ce donne $n = 2^m$ le plus petit entier plus grand que $2d+1$. On doit prendre $n$ points distincts, on choisie $\omega_k = e^{\frac{2i\pi k}{n}}$ pour $k \in \{0, \ldots, n-1\}$. On note $\omega = \omega_1$.

L'algorithme va suivre le schéma suivant :
Entrés $A = (a_0, \ldots, a_{n-1})$ et $A = (b_0, \ldots, b_{n-1})$.
- Évaluation de $(A(1), A(\omega_1), \ldots, A(\omega_{n-1}))$ et $(B(1), B(\omega_1), \ldots, B(\omega_{n-1}))$.
- Calcule de $C(\omega_k) = A(\omega_k)B(\omega_k)$ pour $k \in \{0, \ldots, n-1\}$.
- Interpolation de $C$ avec ces valeurs $(C(1), C(\omega_1), \ldots, C(\omega_{n-1}))$.

Sortie $C = (c_0, \ldots, c_{n-1})$.

**Pour l'évaluation :** On peut voir l'évaluation en terme de système linéaire.

$$\begin{pmatrix} A(\omega_0) \\ \vdots \\ A(\omega_{n-1}) \end{pmatrix} = \begin{pmatrix} 1 & \omega_0 & \ldots & \omega_0^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega_{n-1} & \ldots & \omega_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

On appelle $M_n(\omega)$ cette matrice. Mais on a une manière plus efficace de calculer ce produit matricielle.
Commence par remarquer que $A$ peut ce décomposer sous la forme suivante $A(x) = A_e(x^2) + xA_o(x^2)$.
De plus sous cette forme on a $A(-x) = A_e(x^2) - xA_o(x^2)$.
D'où pour $k \in \{0, \ldots, \frac{n}{2} - 1\}$, on a $A(\omega_k) = A_e(\omega_k^2) + \omega_k A_o(\omega_k^2)$ et
$A(\omega_{k+\frac{n}{2}}) = A(-\omega_k) = A_e(\omega_k^2) - \omega_k A_o(\omega_k^2)$.
Évaluer $A$ de degré $d$ sur $(\omega_k)_{k\in\{0,\ldots,n-1\}}$ se réduit à évaluer $A_e$ et $A_o$ de degré $\frac{d}{2}$ sur $(\omega_{2k})_{k\in\{0,\ldots,\frac{n}{2}-1\}}$.
On en déduit l'algorithme qui suit.

FFT$((a_0,\ldots,a_{n-1}),\omega,n)$ :
si $\omega = 1$, retrouner $a_0$
sinon
$A_e = (a_0, a_2, \ldots, a_{n-2})$
$A_o = (a_1, a_3, \ldots, a_{n-1})$
$y_e = \mathrm{FFT}(A_e, \omega^2, \frac{n}{2})$
$y_o = \mathrm{FFT}(A_o, \omega^2, \frac{n}{2})$
Pour $k = 0$ à $\frac{n}{2} - 1$ faire
$z_k = y_{e,k} + \omega y_{o,k}$
$z_{k+\frac{n}{2}} = y_{e,k} - \omega y_{o,k}$
retourne $(z_0, \ldots, z_{n-1})$
fin

Si on appelle $T$ la complexité de l'algorithme on a donc $T(n) = 2T(\frac{n}{2}) + \mathrm{O}(n)$. C'est à dire $T(2^m) = 2T(2^{m-1}) + \mathrm{O}(2^m)$, ce qui donne $\frac{T(2^m)}{2^m} = \frac{T(2^{m-1})}{2^{m-1}} + \mathrm{O}(1)$ d'où $\frac{T(2^m)}{2^m} = \mathrm{O}(m)$. Finalement on a $T(n) = \mathrm{O}(n\log(n))$.

**Pour la multiplication :** On multiplie terme à terme les vecteurs FFT$((a_0,\ldots,a_{n-1}),\omega,n)$ et FFT$((b_0,\ldots,b_{n-1}),\omega,n)$, ce qui nous donne un vecteur $(C(\omega_0),\ldots,C(\omega_{n-1}))$ en $\mathrm{O}(n)$.

**Pour l'interpolation :** On cherche $(c_0,\ldots,c_{n-1})$ tel que

$$\begin{pmatrix} C(\omega_0) \\ \vdots \\ C(\omega_{n-1}) \end{pmatrix} = M_n(\omega) \begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix}$$

On remarque que $M_n(\omega)^{-1} = \frac{1}{n} M_n(\omega^{-1})$ en effet pour $i,j \in \{0,\ldots,n-1\}$ on a

$$\left(M_n(\omega) M_n(\omega^{-1})\right)_{i,j} = \sum_{k=0}^{n-1} \omega_i^k \omega_k^{-j} = \sum_{k=0}^{n-1} (\omega^{i-j})^k = n\delta_{i,j}$$

Donc $(c_0,\ldots,c_{n-1}) = \frac{1}{n}\mathrm{FFT}((C(\omega_0),\ldots,C(\omega_{n-1})),\omega^{-1},n)$ en $\mathrm{O}(n\log(n))$.

Donc pour finir on a $C = \frac{1}{n}\mathrm{FFT}(\mathrm{Produit}(\mathrm{FFT}(A,\omega,n),\mathrm{FFT}(B,\omega,n)),\omega^{-1},n)$ en $\mathrm{O}(n\log(n))$.