Université Paris Cité
UFR de Mathématiques et Informatique
45, rue des Saints-Pères, 75006, Paris.

License $3^{\text{ème}}$ année, Mathématiques et Applications, 2022-2023
TRANSFORMÉE DE FOURIER ET APPLICATIONS

Feuille de TD n°3

FFT : algorithme de la transformée de Fourier rapide.

Les exercices sont globalement indépendants mais nécessitent parfois des résultats vus dans les précédents. Ils sont pensés pour être faits dans l'ordre.

Exercice 1 (Produit polynomial)
Soit $A = a_0 + a_1X + \ldots + a_nX^n \in \mathbb{C}[X]$ et $B = a_0 + a_1X + \ldots + a_mX^m \in \mathbb{C}[X]$ deux polynômes complexes de degrés respectifs $n \in \mathbb{N}$ et $m \in \mathbb{N}$.

1. a) On note $C = AB$. Montrer que les coefficients de C notés c_ℓ, pour tout $\ell \in \{0, 1, \ldots, n+m\} \in \mathbb{C}^{n+m+1}$, vérifient

$$c_\ell = \sum_{k=0}^{\ell} a_{\ell-k} b_k.$$

b) Pourquoi ne peut-on pas écrire $c = (c_0, c_1, \ldots, c_{n+m})$ comme le produit de convolution $a * b$ où $a = (a_0, a_1, \ldots, a_n)$ et $b = (b_0, b_1, \ldots, b_m)$?

2. On pose $N = n + m + 1$ le nombre de coefficients du polynôme C. On note désormais a, b, c les éléments de ℓ_N définis par

$$(a(0), a(1), \ldots, a(N-1)) = (a_0, a_1, \ldots, a_n, 0, \ldots, 0),$$
$$(b(0), b(1), \ldots, b(N-1)) = (b_0, b_1, \ldots, b_m, 0, \ldots, 0),$$
$$(c(0), c(1), \ldots, c(N-1)) = (c_0, c_1, \ldots, c_{N-1}).$$

a) Montrer cette fois que

$$c = a * b.$$

b) Quelle est la complexité de ce calcul ?

3. On fixe $z_0, z_1, \ldots, z_{N-1}$ des nombres complexes deux à deux distincts.

a) Montrer que $\Phi : \mathbb{C}_{N-1}[X] \to \mathbb{C}^N$ définie par $\Phi(P) = (P(z_0), P(z_1), \ldots, P(z_{N-1}))$, pour tout polynôme P de degré au plus $N-1$, est un isomorphisme d'espaces vectoriels.

b) Montrer alors que $\Phi^{-1}(A(z_0)B(z_0), A(z_1)B(z_1), \ldots, A(z_{N-1})B(z_{N-1})) = AB$. En déduire une méthode alternative à celle présentée en 2)a) pour calculer le produit AB en passant par Φ.

c) Quelle est la complexité du calcul du vecteur $(A(z_0)B(z_0), A(z_1)B(z_1), \ldots, A(z_{N-1})B(z_{N-1}))$ à partir des valeurs $A(z_0), \ldots, A(z_{N-1})$ et $B(z_0), \ldots, B(z_{N-1})$? La méthode de la question précédente utilisée pour calculer le produit AB est-elle plus efficace que celle vue en 2)a) ?

Exercice 2 (DFT et polynôme)
Soit $a_0, a_1, \ldots, a_{N-1} \in \mathbb{C}$ et $a \in \ell_N$ tel que pour tout $k \in \{0, \ldots, N-1\}$, $a(k) = a_k$. On note $A = a_0 + a_1X + \ldots + a_{N-1}X^{N-1} \in \mathbb{C}_{N-1}[X]$ et $\alpha = e^{-\frac{2i\pi}{N}}$.

1. Rappeler la définition de DFT(a).
2. Exprimer DFT(a) en fonction de A et α. Commenter.
3. Retrouver grâce à la question 3)a) de l'exercice 1 que DFT est un isomorphisme d'espaces vectoriels.
4. Quelle est la complexité du calcul de DFT(a) ?

5. En reprenant les notations de l'exercice précédent pour a, b, retrouver dans ce cas particulier que $\mathrm{DFT}(a * b) = \mathrm{DFT}(a) \cdot \mathrm{DFT}(b)$.

Exercice 3 (Fast Fourier Transform)
On suppose que $N = 2^p$ avec $p \in \mathbb{N}$. Soit $a_0, a_1, \ldots, a_{N-1} \in \mathbb{C}$ et $a \in \ell_N$ tel que pour tout $k \in \{0, \ldots, N-1\}$, $a(k) = a_k$. On note $A = a_0 + a_1 X + \ldots + a_{N-1} X^{N-1} \in \mathbb{C}_{N-1}[X]$ et $\alpha = e^{-\frac{2i\pi}{N}}$.

On souhaite trouver une méthode pour calculer $\mathrm{DFT}(a)$ de manière efficace (mieux que $O(N^2)$).

1. Rappeler le lien entre $\mathrm{DFT}(a)$ et A.
2. Montrer qu'il existe A_e et A_o des polynômes de degrés $\frac{N}{2} - 1 = 2^{p-1} - 1$ tels que $A(X) = A_e(X^2) + X A_o(X^2)$. On donnera leurs coefficients respectifs. *Indication : On pourra séparer les monômes composant A en fonction de leur parité.*.
3. Montrer que pour tout $j \in \{0, 1, \ldots, \frac{N}{2} - 1\}$, on a
$$A(\alpha^j) = A_e(\alpha^{2j}) + \alpha^j A_o(\alpha^{2j}),$$
$$A(\alpha^{j+\frac{N}{2}}) = A_e(\alpha^{2j}) - \alpha^j A_o(\alpha^{2j}).$$
4. Que représente l'ensemble $S_{2^p} = \{\alpha^j, \alpha^{j+\frac{N}{2}} : j \in \{0, 1, \ldots, \frac{N}{2} - 1\}\}$?
5. Que représente l'ensemble $S_{2^{p-1}} = \{\alpha^{2j} : j \in \{0, 1, \ldots, \frac{N}{2} - 1\}\}$?
6. Représenter sur le cercle trigonométrique l'ensemble S_{2^3} et mettre en évidence l'ensemble S_{2^2}.
7. Supposons connues les valeurs $A_e(\alpha^{2j})$ et $A_o(\alpha^{2j})$ pour $j \in \{0, 1, \ldots, \frac{N}{2} - 1\}$. Montrer alors que l'on peut déterminer $(A(\alpha^0), A(\alpha), \ldots, A(\alpha^{N-1}))$ en $O(N)$ opérations.
8. On est donc ramené à déterminer $(A_e(\alpha^0), A_e(\alpha^2), \ldots, A_e(\alpha^{2N-2}))$ et $(A_o(\alpha^0), A_o(\alpha^2), \ldots, A_o(\alpha^{2N-2}))$. Expliquer comment on peut de nouveau itérer le procédé précédent. Donner la complexité de cette nouvelle étape.
9. Jusqu'où peut-on répéter cette méthode ? L'ensemble de ces étapes correspond à l'algorithme FFT.
10. Evaluer la complexité totale de l'algorithme FFT.
11. Proposer une méthode pour calculer efficacement un produit polynomial à l'aide de la FFT.