Université Bordeaux 1 | **Algèbre et calcul formel – Master**
Mathématiques | Année 2018–2019

FEUILLE D'EXERCICES n° 5

Travail sur machine

FFT

Durant cette séance, les calculs seront faits en approximation numérique (voir la fonction `numerical_approx()`).

Exercice 1 – [FFT]
Soit n une puissance de 2 différente de 1 : $n = 2^k$ avec $k > 0$. Soit ω une racine primitive n-ième de l'unité, par exemple $\omega = e^{2i\pi/n}$. On rappelle que si P est un polynôme de $\mathbb{C}[X]$ de degré $< n$, que l'on identifiera au n-uplet $(P[0], \ldots, P[n-1])$ on a

$$\mathcal{F}_\omega(P) = (P(1), P(\omega), \ldots, P(\omega^{n-1})).$$

On rappelle aussi que si $P(X) = P_0(X^2) + xP_1(X^2)$, on peut se ramener au calcul de deux transformées de Fourier en degré $< m = n/2$ par le biais des formules

$$\begin{cases} P(\omega^p) & = & P_0(\alpha^p) + \omega^p P_1(\alpha^p) \\ P(\omega^{p+m}) & = & P_0(\alpha^p) - \omega^p P_1(\alpha^p) \end{cases}$$

où $0 \leqslant p < m$ et où $\alpha = \omega^2$.

Rédiger l'algorithme récursif s'appuyant sur cette remarque. La procédure dite `FFT` recevra en entrées P, ω et n, et retournera $\mathcal{F}_\omega(P)$. On prendra garde à ne pas calculer ω^p à chaque étape de la boucle sur p.

Exercice 2 – [Produit rapide de polynômes par FFT]
Ici encore $n = 2^k$ avec $k > 0$. Soient P et Q deux polynômes de $\mathbb{C}[X]$ vérifiant $\deg(PQ) < n$ (on pourra imposer que $\deg P, \deg Q < n/2$). On identifiera encore P et Q aux n-uplets $(P[0], \ldots, P[n-1])$ et $(Q[0], \ldots, Q[n-1])$. On rappelle que l'on a alors

$$\mathcal{F}_\omega(PQ) = \mathcal{F}_\omega(P) \cdot \mathcal{F}_\omega(Q)\,[1],$$

et que pour tout polynôme $R \in \mathbb{C}[X]$ de degré $< n$ on a

$$\mathcal{F}_{\omega^{-1}}(\mathcal{F}_\omega(R)) = \mathcal{F}_\omega(\mathcal{F}_{\omega^{-1}}(R)) = nR.$$

Écrire une procédure prenant en arguments P, Q et n et retournant PQ, procédure qui prendra bien sûr appui sur la procédure `FFT` de l'exercice 1.

Expérimenter cette fonction sur des polynômes pris au hasard.

1. ici $(u_i)_{0\leqslant i<n} \cdot (v_i)_{0\leqslant i<n} = (u_i v_i)_{0\leqslant i<n}$.

Exercice 3 – [La fonction `fft` de sage]

1) Expérimenter les commandes suivantes.

```
A=[RR(1) for i in range(8)]
s=IndexedSequence(A,range(8))
t=s.fft();t
lt=t.list();lt
```

2) Comparer les résultats donnés par cette fonction `fft` avec les résultats de l'exercice 1.

Exercice 4 – [Filtres]

Soit f une fonction périodique de période 1 de $\mathbb{R}$ dans $\mathbb{C}$. Il suffit de connaître f sur $[0,1]$ pour connaître f. Soient N une puissance de 2 et $w = e^{2i\pi/N}$. Soit

$$F = \left[f(0), f\left(\frac{1}{N}\right), \dots, f\left(\frac{N-1}{N}\right)\right].$$

Pour tout n,

$$F[n] = \frac{1}{N}\sum_{k=0}^{N-1} \mathcal{F}_{w^{-1}}(F)[k]w^{nk}.$$

Plus simplement, si l'on note $\hat{F} = \dfrac{1}{N}\mathcal{F}_{w^{-1}}(F)$, alors

$$F[n] = \sum_{k=0}^{N-1} \hat{F}[k]w^{nk}. \qquad (1)$$

Ainsi, la transformation $F \mapsto \hat{F}$ permet de passer de F à la suite des composantes fréquencielles de F (voir aussi la question qui suit).

1) Soit pour $k \in [[0, n-1]]$ $W_k = [1, \omega^k, \dots, \omega^{(n-1)k}]$. Soit $< \cdot, \cdot >$ le produit scalaire hermitien défini sur $\mathbb{C}^n$ par $< x, y >= \sum_{k=1}^{n-1} \overline{x}_i y_i$. Montrer que $(W_k)_{k\in[[0,n-1]]}$ est une base orthogonale de $\mathbb{C}^n$.

L'équation (1) s'écrit aussi $F = \sum_{k=0}^{N-1} \hat{F}[k]W_k$: les $\hat{F}[k]$ sont les coordonnées de F dans la base (W_k).

2) Étudions un exemple simple. Soit $f(t) = \sin(18\pi t) + 3\cos(10\pi t)$. Soient $N = 1024$ et $w = e^{i\pi/512}$. On définit F comme ci-dessus.

a) En utilisant votre fonction pour la fft, calculer $\hat{F}$.

b) Représenter sur un graphique la ligne brisée définie par les points

$$\left[(0, F[0]), \dots, \left(\frac{1023}{1024}, F[1023]\right)\right]$$

(on pourra utiliser la fonction `line`).

Représenter sur un autre graphique les points

$$\left[(0,|\hat{F}[0]|),\ldots,\left(\frac{1023}{1024},|\hat{F}[1023]|\right)\right]$$

en utilisant la fonction `point`. Pour le module, on utilise `abs`. Commenter ce graphique.

c) Calculer $\mathcal{F}_w(\hat{F})$, et représenter sur un graphique la ligne brisée définie par cette fonction (attention : comme les coefficients trouvés sont des valeurs approchées dans $\mathbb{C}$, il faudra prendre les parties réelles). Comparer cette ligne brisée avec la ligne brisée correspondant à F.

Pour mieux les comparer, on peut les placer sur le même graphique en s'inspirant du modèle suivant, où A et B sont des listes, et où x est la liste des $\dfrac{i}{1024}$.

```
GrapheA=line([(x[i],A[i]) for i in range(1024)],color='red')
GrapheB=line([(x[i],B[i]) for i in range(1024)])
GrapheA+GrapheB
```

Les deux premières lignes sont des affectations. La troisième trace les deux lignes sur un même graphique. Cela peut aussi s'écrire

```
GrapheA=line(zip(x,A),color='red')
GrapheB=line(zip(x,B))
GrapheA+GrapheB
```

d) Introduisons un bruit dans le signal F.

```
def h():
    return ZZ.random_element(-500,500)/1000
```

définit une fonction aléatoire. Soit le bruit

```
B=[h() for i in range(1024)]
```

Ajoutons ce bruit à F.

```
FB=[F(i)+B[i] for i in range(1024)]
```

Faire un graphe dessinant la ligne définie par FB et la comparer à celle de F. On suppose qu'au lieu du signal F, on ait reçu le signal brouillé FB. On va essayer de reconstituer F, en considérant que les fréquences ajoutée par le bruit sont de faible amplitude.

e) Calculer $\widehat{FB}$ et dessiner l'ensemble des points correspondants. Comparer avec les points donnés par $\hat{F}$.

f) Soit G définie par : $G[i]=\widehat{FB}[i]$ si $|\widehat{FB}[i]|>1/4$ et $G[i]=0$ sinon. Calculer $\mathcal{F}_w(G)$. Imprimer son graphe et comparer avec F. Commenter.

3) Soit f la fonction périodique de période 1 définie sur $[0,1[$ par

$$f(t)=1920(t-1/6)(t^2-1/2)(t-1/2)(t-7/8)(t-1/10)(t-1)t.$$

Refaire l'exercice précédent sur cette fonction, en prenant toujours $N=1024$ et $w=e^{2i\pi/1024}$. Quand on filtre le signal brouillé (question 2.f), on peut modifier le seuil à partir duquel on met $\widehat{FB}[k]$ à 0.

4) On peut aussi choisir de filtrer les fréquences situées dans certains intervalles. Reprendre la question 3 en mettant à 0 les $\widehat{FB}[k]$ tels que $k \in [[11, 1013]]$.

5) Expérimenter la fonction définie sur $[0, 1[$ par

$$f(t) = 1920(t - 1/6)(t^2 - 1/2)(t - 1/2)(t - 7/8)(t - 1/10)(t - 1).$$

Exercice 5 – [Inverse de la transformée de Fourier discrète]

Démontrer le résultat du cours suivant. Soient K un corps contenant une racine primitive n-ème de 1 notée ω. Alors $\mathcal{F}_\omega \circ \mathcal{F}_{\omega^{-1}} = n \cdot \mathrm{Id}$.

Travail sur papier

Suites de Fibonacci et algorithme d'Euclide

Exercice 6 – [Suite de Fibonacci et nombre d'or]

Soient $\Phi = \frac{1+\sqrt{5}}{2}$ le nombre d'or et $\Phi' = \frac{1-\sqrt{5}}{2}$ son conjugué. Soit (F_n) la suite de Fibonacci définie par $F_0 = 0$, $F_1 = 1$ et $F_{n+1} = F_n + F_{n-1}$ pour tout $n \geqslant 1$. Montrer que pour tout $n \geqslant 0$,

$$F_n = \frac{\Phi^n - \Phi'^n}{\sqrt{5}}$$

En déduire que l'ordre de grandeur de la suite F_n est exponentielle.

Exercice 7 – [Divisions dans l'algorithme d'Euclide]

On applique l'algorithme d'Euclide aux entiers a et b tels que $a > b$. Soit t le nombre de divisions nécessaires. On reprend les notations de l'exercice 6.

1) Montrer que $a \geqslant F_{t+2}$ et que l'algorithme d'Euclide appliqué à $a = F_{t+2}$ et $b = F_{t+1}$ demande t divisions.

2) On a vu que $t \leqslant \lfloor \log_\Phi(a) \rfloor$. Montrer que l'égalité est atteinte si $a = F_{t+2}$ et $b = F_{t+1}$, c'est à dire, montrer que $t = \lfloor \log_\Phi(F_{t+2}) \rfloor$.

Exercice 8 – [Raccourci]

On reprend a, b et t comme dans l'exercice 7. On va évaluer l'ordre de grandeur de t de façon plus grossière que ce qui a été fait en cours. C'est un peu plus facile et c'est suffisant pour l'ordre de grandeur de la complexité de l'algorithme mais on n'y met pas en évidence le lien avec la suite de Fibonacci.

Soient $r_0 = a$, $r_1 = b$ et pour tout i, on note $r_{i-1} = r_i q_i + r_{i+1}$ la division euclidienne de r_{i-1} par r_i.

1) Montrer que $r_{i+1} < r_{i-1}/2$.

2) En déduire que $r_{2i} < \dfrac{a}{2^i}$ pour tout $i > 0$ et que $t < 2\log_2(a)$.