Analyse de Fourier et hilbertienne
ENS de Cachan
Département de Mathématiques Vendredi 29 Novembre 2013

Feuille de TP numéro 2

1 Consignes et mise en route

1.1 Rapport de TP

- Consignes pour le rapport de TP : cette feuille de TP comporte des questions auxquelles il est demandé de répondre dans un rapport de TP individuel.

- Format du rapport : le rapport doit être rédigé en LaTeXet rendu au format `.pdf`.

- Remise du rapport : le rapport de ce TP doit être rendu avant le **Vendredi 13 Decembre 2013**. Les rapports doivent être envoyés par email à l'adresse **gabriele.facciolo@cmla.ens-cachan.fr**. Le sujet de l'email doit être **Rapport du TP2 de *Nom Prénom*** et le fichier `.pdf` doit être nommé ***nom_prenom*_tp2.pdf**.

1.2 Téléchargement des fichiers nécessaires

Dans un terminal, tapez les instructions suivantes pour télécharger les fichiers nécessaires et lancer Matlab :

```
mkdir -p tp_fourier
cd tp_fourier
wget dev.ipol.im/~facciolo/hilbert/tp2.tar.gz
tar xvfz tp2.tar.gz
cd tp2
matlab
```

2 Rappels : Images numériques et polynômes trigonométriques

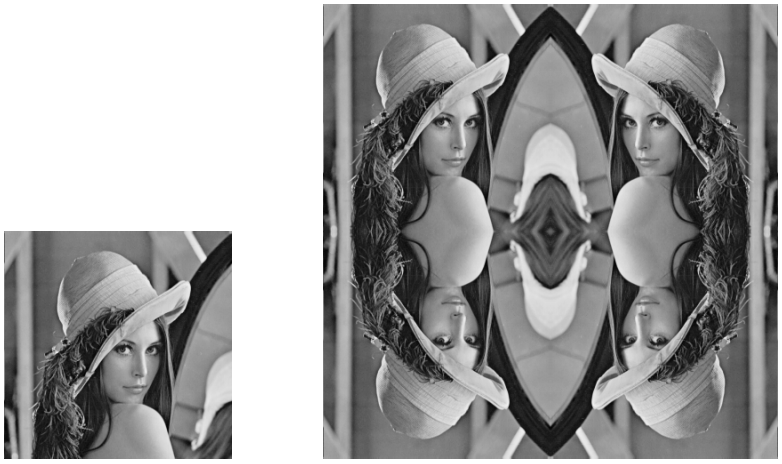
Dans tout le TP on considère une image numérique

$$u = (u_{m,n}), \quad (m,n) \in [\![0, M-1]\!] \times [\![0, N-1]\!],$$

où **les entiers M et N sont supposés pairs**. La transformée de Fourier discrète (TFD) de u est notée $\tilde{u} = (\tilde{u}_{m,n})$, avec $(m,n) \in [\![-\frac{M}{2}, \frac{M}{2}-1]\!] \times [\![-\frac{N}{2}, \frac{N}{2}-1]\!]$. On rappelle que le polynôme trigonométrique (M,N)-périodique P_u associé à u est défini par

$$P_u(x,y) = \sum_{m=-\frac{M}{2}}^{\frac{M}{2}-1} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_{m,n} e^{\frac{2i\pi mx}{M}} e^{\frac{2i\pi ny}{N}}. \tag{1}$$

Ce TP vise à résoudre de manière exacte des équations aux dérivées partielles (EDP) dans l'espace des polynômes trigonométriques.



(a) Image originale (b) Image symétrisée

FIGURE 1 – Illustration de la symétrisation miroir qui doit être produite par la fonction `miroir`.

3 Calcul des dérivées d'une image

Un prérequis essentiel pour la résolution numérique d'EDP est de disposer d'algorithmes de calcul des dérivées des images. Ici nous allons identifier l'image u avec son polynôme trigonométrique associé P_u. Nous allons donc implémenter des algorithmes qui calculent exactement les dérivées de P_u évaluées aux points (m, n).

3.1 Gradient d'une image

► **Question 1 :** En dérivant l'expression (1) exprimez les coefficients des TFD des images $\frac{\partial P_u}{\partial x}(m,n)$ et $\frac{\partial P_u}{\partial y}(m,n)$, $(m,n) \in [\![0, M-1]\!] \times [\![0, N-1]\!]$, en fonction de la TFD $\tilde{u}$ de u.

► **Question 2 :** Etudiez la fonction Matlab `[dxu, dyu]=gradient_tfd2(u)`. Vérifiez qu'elle calcule les échantillons des dérivées partielles $\frac{\partial P_u}{\partial x}(x,y)$ et $\frac{\partial P_u}{\partial y}(x,y)$. Testez cette fonction. Visualisez (`imagesc` et `colormap gray`, *cf* TP1) l'image originale, ses deux dérivées partielles, ainsi que la norme du gradient en chaque point. On prendra soin de visualiser la partie réelle du gradient du polynôme trigonométrique (qui a une composante imaginaire, *cf* TP1).

Ainsi calculé, le gradient est sujet aux effets de Gibbs, notamment au bord de l'image. Pour éviter ce problème on utilise une symétrisation préalable de l'image, en la rendant paire selon chaque variable (symétrisation miroir).

► **Question 3 :** Coder la fonction `function v = miroir(u)` (en créant un fichier `miroir.m`) qui à partir d'un image u de taille $M \times N$ calcule l'image v de taille $2M \times 2N$ obtenue par la symétrisation miroir illustrée par la Figure 1. Tester cette fonction sur des images de taille rectangulaire (et pas seulement carré).

► **Question 4 :** A l'aide de la fonction `gradient_tfd2` calculez le gradient de l'image symétrisé `v = mirroir(u)`. Extrayez de ce gradient le sous-domaine correspondant à l'im-

age originale. Comparez visuellement et numériquement ce "gradient symétrique" avec le "gradient périodique" calculé à la question précédente.

3.2 Laplacien d'une image

On s'intéresse maintenant au calcul du Laplacien d'une image en suivant la même démarche qu'à la section précédente.

► **Question 5 :** Donnez la relation entre les coefficients de la TFD de u et ceux de la TFD de

$$\Delta P_u(m,n) = \frac{\partial^2 P_u}{\partial x^2}(m,n) + \frac{\partial^2 P_u}{\partial y^2}(m,n),\ (m,n) \in [\![0, M-1]\!] \times [\![0, N-1]\!].$$

Comme pour le gradient nous allons implémenter deux algorithmes pour le calcul du Laplacien. Le premier va calculer le "Laplacien périodique" $\Delta_{\text{per}} u = \Delta P_u(m,n)$ qui est égal au Laplacien de P_u. Le deuxième va calculer le "Laplacien symétrique" $\Delta_{\text{sym}} u$, qui est égal à la restriction sur le domaine $[\![0, M-1]\!] \times [\![0, N-1]\!]$ du Laplacien périodique de l'image symétrisée miroir de u. Plus précisément, si $\mathcal{M}(u)$ est l'image symétrisée de taille $2M \times 2N$ alors

$$\Delta_{\text{sym}} u(m,n) = \Delta_{\text{per}}\left(\mathcal{M}(u)\right)(m,n),\ (m,n) \in [\![0, M-1]\!] \times [\![0, N-1]\!].$$

► **Question 6 :** Ecrivez une fonction `v=laplacien_per_dft2(u)` qui calcule le Laplacien périodique de u. Ecrivez ensuite une fonction `v=laplacien_sym_dft2(u)` qui calcule le Laplacien symétrique de u. Visualisez sur un exemple l'image originale, son Laplacien périodique, son Laplacien symétrique ainsi que la différence de ces deux variantes du Laplacien.

Remarquez que l'opération $u \mapsto \Delta_{\text{per}} u$ est presque inversible puisque, à l'exception du coefficient $\tilde{u}_{0,0}$, chaque coefficient de Fourier est multiplié par un facteur non nul. Ainsi seul le coefficient de Fourier $\tilde{u}_{0,0}$ n'est pas déterminé par la donnée de $\Delta_{\text{per}} u$ (on rappelle que $\tilde{u}_{0,0}$ représente soit la moyenne de l'image u soit la somme de tous ses coefficients, selon la convention TFD/TFD inverse).

Remarquez également que les opérateurs différentiels dépendent des conditions au bord. Ceci est important en pratique. Par exemple, pour résoudre l'équation de Poisson avec des conditions périodiques on doit utiliser l'opérateur Δ_{per}. En revanche, pour résoudre l'équation de Poisson avec des conditions de Neumann (bord adiabatique) on doit utiliser l'opérateur Δ_{sym}. Pour les conditions de Dirichlet (bord isotherme), il faut utiliser un troisième Laplacien obtenu à partir d'une symétrisation impaire de l'image.

4 Composante périodique d'une image

Dans cette section on définit et calcule la composante périodique d'une image u. La composante périodique d'une image répond au problème suivant : on cherche à déterminer une image $p = \text{per}(u)$ qui soit à la fois proche de u et qui soit "analytiquement périodique", c'est-à-dire avec de faibles sauts d'un bord à l'autre de l'image.

Définition 1. *(Composante périodique) Soit u une image numérique. La composante périodique $p = \text{per}(u)$ est l'unique solution p du problème de Poisson suivant :*

$$\begin{cases} \Delta_{\text{per}} p = \Delta_{\text{sym}} u, \\ \text{moy}(p) = \text{moy}(u), \end{cases} \tag{2}$$

où $\mathrm{moy}(u)$ *désigne la moyenne de* u*.*

Le calcul de la composante périodique de u s'effectue donc de la manière suivante.

1. Calculer $\Delta_{\mathrm{sym}} u$.
2. Calculer la TFD de $\Delta_{\mathrm{sym}} u$.
3. Diviser chaque coefficients de Fourier par le facteur multiplicatif qui intervient dans le calcul de $\Delta_{\mathrm{per}} u$, à l'exception du coefficient d'indice $(0,0)$.
4. En $(0,0)$ imposer la valeur $\tilde{u}_{(0,0)}$ (sans toutefois calculer la TFD de u !) pour que la TFD inverse ait la même moyenne que u.
5. Effectuer une TFD inverse.

► **Question 7 :** Ecrivez une fonction `function p=composante_periodique(u)` qui calcule la composante périodique d'une image u.

► **Question 8 :** Calculez la composante périodique p de l'image u =`nuages_ng.png`, ainsi que la différence $s = p - u$. Sur une même figure Matlab, visualisez les trois images u, p et s ainsi que leurs modules respectifs (utilisez les commandes `subplot(2,3,k)` et la fonction `vis_mod_img(v)`). Commentez.

5 Images couleur

Dans la suite du TP nous allons utiliser des images numériques en couleur. Une image couleur u est représentée par trois images en niveau de gris $u = (u_r, u_g, u_b)$. Chacune de ces trois images représente la quantité de rouge, de vert et de bleu en chaque pixel. On parle alors de représentation RGB (pour red, green, blue).

Pour Matlab, une image couleur est donc un tableau de chiffre à trois dimension de taille $M \times N \times 3$. Pour visualiser une image couleur elle doit être à valeurs entieres et dans l'intervalle $[0, 255]$. Donc on utilisera `imagesc(uint8(u),[0,255])` pour la montrer, ou `uint8` convertit les valeurs réelles à entieres. Si les valeurs de `u` ne sont pas dans l'intervalle $[0, 255]$, on utilisera la fonction `imagesc(img_affine(u),[0,255])` appliquant un changement de variable affine à l'image `u` pour se ramener l'intervalle `[min(u(:)),max(u(:))]` sur l'intervalle $[0, 255]$.

A titre d'illustration, **analysez** et exécutez le script intitulé `canaux_couleurs.m` qui lit l'image couleur `singe.png` et affiche sur une même figure l'image couleur ainsi que ses trois canaux couleurs.

6 Synthèse d'images par phase aléatoire

6.1 Bruit à phase aléatoire pour la synthèse de textures

On a observé lors du TP1 que l'information des contours géométriques d'une image est principalement contenue dans la phase de l'image. Inversement, l'information de type texture d'une image est plutôt contenue dans le module de Fourier de l'image.

Cette dernière observation peut être mise à profit pour synthétiser des textures. Le bruit à phase aléatoire (BPA) associé à une image u est une image aléatoire obtenue en remplaçant la

phase de la TFD de u par une phase aléatoire. Comme la phase est rendue incohérente, le BPA transforme n'importe quelle image u en une image de texture. L'intérêt de ce procédé est qu'il permet de resynthétiser certaines textures naturelles.

► **Question 9 :** Calculez à l'aide de la fonction `bpa_couleur.m` une réalisation des BPA associés aux images `singe.png`, `mur.png`, `tissu.png`, `nuages.png`, et `briques.png`. Parmi les quatre images de texture quelles sont celles qui semblent être bien reproduites ?

Pour la question suivante nous avons besoin d'étendre la notion de composante périodique aux images couleurs. La composante périodique d'une image couleur $u = (u_r, u_g, u_b)$ est naturellement définie comme étant l'image couleur p dont les canaux sont les composantes périodiques des canaux de u, c'est-à-dire $p = (\mathrm{per}(u_r), \mathrm{per}(u_g), \mathrm{per}(u_b))$.

► **Question 10 :** Comparez le BPA associé à `nuages.png` et le BPA associé à la composante périodique de `nuages.png`. Comment expliquez-vous les différences observées ? On pourra se référer aux modules de Fourier observés à la question 8 pour la version niveau de gris de l'image `nuages.png`.

6.2 "Fonctions typiques" d'un espace de Sobolev

Dans cette partie on synthétise des images qui représentent des "fonctions typiques" d'espace de Sobolev. On rappelle d'abord le résultat suivant (Proposition 7.6 p. 113 du polycopié).

Proposition 2. *(norme sur les espaces H^k_{per})*
Les espaces $H^k_{\mathrm{per}}([0,2\pi]^2)$ sont des espaces de Hilbert et ont pour norme équivalente

$$\|f\|^2_{H^k_{\mathrm{per}}} = \sum_{(m,n)} |c_{(m,n)}(f)|^2 (1+m^2+n^2)^k.$$

Les coefficients de Fourier des fonctions $f \in H^k_{\mathrm{per}}([0,2\pi]^2)$ décroissent donc rapidement. Dans ces conditions, les coefficients de la TFD des images discrètes $f_N : (m,n) \mapsto f\left(\frac{2\pi m}{N}, \frac{2\pi m}{N}\right)$ sont de bonnes approximations des coefficients de Fourier de f (en vertu du théorème 6.1 p. 89).

► **Question 11 :** Ecrivez une fonction `function u = sobolev(alpha,N)` qui calcule une image u de taille $N \times N$ dont les modules des coefficients de Fourier sont

$$|\tilde{u}_{m,n}| = \left(1+m^2+n^2\right)^{-\frac{\alpha}{2}}$$

et dont la phase est aléatoire (utilisez la fonction `phase_aleatoire` de l'archive). Choisissez trois valeurs de α pour visualiser une fonction f_1 en dehors de $H^1_{\mathrm{per}}([0,2\pi]^2)$, une fonction f_2 dans $H^1_{\mathrm{per}}([0,2\pi]^2) \setminus H^2_{\mathrm{per}}([0,2\pi]^2)$, et une fonction f_3 de $H^2_{\mathrm{per}}([0,2\pi]^2)$. Justifiez les valeurs choisies pour les trois indices α et commentez l'aspect visuel des images synthétisées.

Comme la phase est aléatoire et que la décroissance des coefficients est choisie de manière optimale, on peut argumenter que ces fonctions représentent la "régularité typique" de leur espace de Sobolev respectif.

Enfin du point de vue de la synthèse d'images, remarquez que la fonction `sobolev` peut-être considérée comme un synthétiseur de nuages plus ou moins granuleux selon le paramètre α.