

## Feuille de TP numéro 3

### 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.
- **Barème :** la moyenne des notes obtenues pour les rapports des quatre TP contribuera à la note de partiel à hauteur de 6 points sur 20.
- **Format du rapport :** les rapports de TP sont à rendre sous forme de fichier .pdf rédigés avec  $\text{\LaTeX}$ .
- **Remise du rapport :** le rapport de ce TP doit être impérativement rendu avant le **vendredi 18 décembre 2009**. Les rapports, au format .pdf, doivent être envoyés par email à l'adresse [galerie@cmla.ens-cachan.fr](mailto:galerie@cmla.ens-cachan.fr). Le sujet de l'email doit être **Rapport du TP3 de Nom Prénom** et le fichier .pdf doit être intitulé *nom\_prenom\_tp3*.

#### 1.2 Téléchargement des fichiers nécessaires

Ouvrez un terminal et tapez y ligne par ligne les instructions suivantes pour télécharger les fichiers nécessaires et lancer Matlab (on suppose ici que le dossier `tp_fourier` a été créé dans votre dossier `home` lors du premier TP) :

```
cd tp_fourier
wget mw.cmla.ens-cachan.fr/~galerie/tp3.zip
unzip tp3.zip
cd tp3
matlab
```

### 2 Images numériques et polynômes trigonométriques

Dans tout le TP on considère une image numérique  $u = (u_{m,n})$ ,  $(m, n) \in \llbracket 0, M-1 \rrbracket \times \llbracket 0, N-1 \rrbracket$ , 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  $(m, n) \in \llbracket -\frac{M}{2}, \frac{M}{2}-1 \rrbracket \times \llbracket -\frac{N}{2}, \frac{N}{2}-1 \rrbracket$ . On rappelle que le polynôme trigonométrique  $(M, N)$ -périodique  $P_u$  associé à  $u$  est défini par

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

Le but des TP3 et TP4 est de résoudre de manière exacte des équations aux dérivées partielles (EDP) dans l'espace des polynômes trigonométriques. D'un point de vue pratique il est important de se restreindre à des polynômes trigonométriques à valeur réelle. La proposition suivante explicite une condition nécessaire et suffisante sur  $u$  pour que le polynôme trigonométrique  $P_u$  soit à valeur réelle (cette proposition est l'équivalent en deux dimension de la proposition 6.4 p. 87 du polycopié).

**Proposition 1.** (CNS pour que  $P_u$  soit réel)

Le polynôme trigonométrique  $P_u$  est à valeur réelle sur  $\mathbb{R}^2$  si et seulement si

$$\forall m \in \left[ \left[ -\frac{M}{2}, \frac{M}{2} - 1 \right] \right], \tilde{u}_{m, -\frac{N}{2}} = 0 \quad \text{et} \quad \forall n \in \left[ \left[ -\frac{N}{2}, \frac{N}{2} - 1 \right] \right], \tilde{u}_{-\frac{M}{2}, n} = 0.$$

- **Question 1 :** Ecrivez une fonction Matlab intitulée `function v = annuler_hf(u)` qui prend en entrée une image  $u$  de taille  $M \times N$  et qui retourne l'image  $v$  obtenue en annulant les  $M + N - 1$  coefficients hautes fréquences de la TFD de  $u$ , de telle sorte que le polynôme trigonométrique associé à  $v$  soit à valeur réelle. Testez cette fonction sur l'image `lena.png` (ou une autre image en niveau de gris) et vérifiez que la différence entre  $u$  et  $v$  n'est pas ou peu perceptible. Visualisez également la différence  $u - v$  et commentez l'allure de cette fonction.

### 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  $P_u$  aux points  $(m, n)$ .

#### 3.1 Gradient d'une image

- **Question 2 :** En dérivant l'expression (1) exprimez les coefficients des TFD des images  $\frac{\partial P_u}{\partial x}(m, n)$  et  $\frac{\partial P_u}{\partial y}(m, n)$ ,  $(m, n) \in \llbracket 0, M - 1 \rrbracket \times \llbracket 0, N - 1 \rrbracket$ , en fonction de la TFD  $\tilde{u}$  de  $u$ .
- **Question 3 :** Ecrivez une fonction Matlab `function [dxu, dyu]=gradient_tfd2(u)` qui calcule les images  $\frac{\partial P_u}{\partial x}(m, n)$  et  $\frac{\partial P_u}{\partial y}(m, n)$ . Testez cette fonction sur une image sans hautes fréquences et visualisez l'image originale, les deux dérivées partielles, ainsi que la norme du gradient en chaque point.

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). Cette symétrie est réalisée par la fonction `miroir` (voir le fichier `miroir.m`).

- **Question 4 :** A l'aide de votre fonction `gradient_tfd2` calculez le gradient de l'image symétrisé  $v = \text{miroir}(u)$ . Extrayez de ce gradient le sous-domaine correspondant à l'image originale. Comparez 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

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

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 égale au laplacien de  $P_u$ . Le deuxième va calculer le "laplacien symétrique"  $\Delta_{\text{sym}}u$ , qui est égale à la restriction sur le domaine  $\llbracket 0, M-1 \rrbracket \times \llbracket 0, N-1 \rrbracket$  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

$$\Delta_{\text{sym}}u(m, n) = \Delta_{\text{per}}(\mathcal{M}(u))(m, n), \quad (m, n) \in \llbracket 0, M-1 \rrbracket \times \llbracket 0, N-1 \rrbracket.$$

- **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 des deux laplaciens.

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 de bord. Ceci a son importance en pratique. Par exemple, pour résoudre l'équation de Poisson avec des conditions périodiques on doit utiliser l'opérateur  $\Delta_{\text{per}}$ . En revanche, pour résoudre l'équation de Poisson avec des conditions de Neumann (bord adiabatique) on doit utiliser l'opérateur  $\Delta_{\text{sym}}$ . 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 2.** (composante périodique)

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

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

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

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

1. Calculer  $\Delta_{\text{sym}}u$ .
2. Calculer la TFD de  $\Delta_{\text{sym}}u$ .
3. Diviser chaque coefficients de Fourier par le facteur multiplicatif qui intervient dans le calcul de  $\Delta_{\text{per}}u$ , 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 `fonction p=composante_periodique(u)` qui calcule la composante périodique d'une image (sans hautes fréquences)  $u$ .

► **Question 8 :** Calculez la composante périodique  $p$  de l'image  $u = \text{nuages\_ng.png}$ , 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 on utilisera les fonctions `vis_imgc_mat` et `vis_imgc_ext`.

A titre d'illustration, 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 d'une image. Inversement, l'information de la texture d'une image est plutôt contenu dans le module de Fourier d'une 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 obtenu en remplaçant la phase de la TFD de  $u$  par une phase aléatoire. Comme la phase est rendue incohérente, le BPA transforme n'importe qu'elle image  $u$  en une texture. L'intérêt de ce procédé est qu'il est capable 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  $p = (\text{per}(u_r), \text{per}(u_g), \text{per}(u_b))$ .

- **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 ?

## 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 3.** (norme sur les espaces  $H_{\text{per}}^k$ )

Les espaces  $H_{\text{per}}^k([0, 2\pi]^2)$  sont des espaces de Hilbert et ont pour norme équivalente

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

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

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

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

et dont la phase est aléatoire (utilisez la fonction `phase_aleatoire`). Choisissez trois valeurs de  $\alpha$  pour visualiser une fonction  $f_1$  en dehors de  $H_{\text{per}}^1([0, 2\pi]^2)$ , une fonction  $f_2$  dans  $H_{\text{per}}^1([0, 2\pi]^2) \setminus H_{\text{per}}^2([0, 2\pi]^2)$ , et une fonction  $f_3$  de  $H_{\text{per}}^2([0, 2\pi]^2)$ .

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 leurs espaces de Sobolev respectifs.

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  $\alpha$ .