Exercice 6 (sinus cardinal discret)

Développer en série de Fourier la fonction f 2π-périodique définie par $f(x) = \cos(\alpha x)$ pour $x \in [-\pi, \pi]$ ($\alpha \in \mathbb{C} \setminus \mathbb{Z}$). En déduire les développements eulériens

$$\frac{1}{\tan t} = \frac{1}{t} + \sum_{n=1}^{\infty} \frac{2t}{t^2 - n^2\pi^2}, \quad \frac{1}{\sin t} = \frac{1}{t} + \sum_{n=1}^{\infty} (-1)^n \frac{2t}{t^2 - n^2\pi^2}.$$

En déduire, en fonction de la parité de l'entier N, la valeur de

$$\lim_{K \to +\infty} \sum_{k=-K}^{K} \operatorname{sinc}(x - kN),$$

et faire le lien avec le cours.

Exercice 7 (transformée de Fourier)

La commande matlab/octave fft2 permet de calculer la transformée de Fourier discrète d'un signal fini de dimension quelconque (pour nous de dimension 2). Commentez les résultats donnés par les commandes suivantes (la fonction `normsat` est à récupérer dans la toolbox `iptools` du cours) :

```
u = double(imread('lena.pgm'));
f = fft2(u);
imshow(f);
imshow(abs(f));
imshow(abs(f),[]);
imshow(normsat(abs(f),1));
imshow(normsat(fftshift(abs(f)),1));
```

Proposez des commandes pour visualiser efficacement la partie réelle de f, la partie imaginaire de f, la phase (argument) de f, puis le module de f en échelle logarithmique. Pour la phase, on pourra utiliser la fonction `angle` (voir l'aide: `help angle`).

Exercice 8 (périodisation implicite de la transformée de Fourier discrète)

1) La transformée de Fourier discrète manipule implicitement une image discrète comme une image périodique. Une manière de mettre en évidence ce phénomène est d'expliciter le lien entre la transformée de Fourier discrète d'une image $u : \Omega \to \mathbb{R}$ ($\Omega = \{0, ..., M-1\} \times \{0, ..., N-1\}$), et la transformée de Fourier de sa distribution périodique associée

$$U = \sum_{0 \le k \le M-1,\, 0 \le l \le N-1} u(k, l) \sum_{m,n \in \mathbb{Z}} \delta_{(k+mM, l+nN)}.$$

Retrouver par le calcul le lien entre $\hat{u}$ et $\hat{U}$.

2) Une autre façon de se convaincre de ce phénomène de périodisation implicite est de comparer la transformée de Fourier d'une image discrète $u : \Omega \to \mathbb{R}$ et de sa translatée périodique $u' : \Omega \to \mathbb{R}$ définie par

$$u'(k,l) = \dot{u}(k + k_0, l + l_0),$$

où $\dot{u}$ est l'extension périodique de u à $\mathbb{Z}^2$ et k_0, l_0 deux entiers quelconques. Quelle est la relation entre $\hat{u}$ et $\widehat{u'}$?

3) Retrouvons maintenant le résultat de la question 3 en simulant numériquement une telle translation au moyen de la fonction `fshift` de la toolbox `iptools` (vous pouvez également utiliser, si vous disposez de la toolbox *Image Processing* de Matlab, la fonction `imtranslate`, avec les bons arguments pour avoir une translation périodique) :

```
u = double(imread('lena.pgm'));
imshow(u,[]);
v = fshift(u,-30,-30);
figure();imshow(v,[]);
```

Comparer ensuite les transformées de Fourier des images `u` et `v` en visualisant leur module (en échelle log) et leur phase (argument).

Exercice 9 (interpolation de Shannon avec partie réelle)

On considère une image discrète $u : \Omega \to \mathbb{R}$, où $\Omega = \{0, ..., M-1\} \times \{0, ..., N-1\}$ et les deux entiers M et N sont pairs. Écrire l'interpolée de Shannon $v(x,y)$ en fonction de $\hat{u}$, la transformée de Fourier discrète de u. On considère la fonction $w : \mathbb{R}^2 \to \mathbb{R}$ définie par

$$w(x,y) = \mathrm{Re}\left(\frac{1}{MN} \sum_{\substack{-\frac{M}{2} \leq \alpha < \frac{M}{2} \\ -\frac{N}{2} \leq \beta < \frac{N}{2}}} \widehat{u}(\alpha,\beta)\, e^{2i\pi\left(\frac{\alpha x}{M} + \frac{\beta y}{N}\right)} \right).$$

Montrer que w définit, comme v, une interpolation exacte de u et calculer explicitement $w - v$ (on prendra soin d'aboutir à l'expression la plus simple possible, en exploitant les relations entre les coefficients de $\hat{u}$). Quelle interpolation est la plus satisfaisante du point de vue des invariances géométriques, celle donnée par v ou par w ?

Exercice 10 (synthèse de microtextures)

1) Comment appréhender la signification des coefficients de Fourier d'une image, et en particulier les rôles respectifs de la phase et du module de ces coefficients ? Une manipulation intéressante que l'on peut effectuer à partir de deux images est simplement d'échanger les phases et les modules de leurs coefficients de Fourier. Ainsi, si l'on écrit

$$\hat{u}_1(p,q) = \rho_1(p,q) e^{i\varphi_1(p,q)} \qquad \text{et} \qquad \hat{u}_2(p,q) = \rho_2(p,q) e^{i\varphi_2(p,q)}$$

(u_1 et u_2 étant deux images de même taille), on peut alors produire artificiellement deux nouvelles images u_1' et u_2' vérifiant

$$\hat{u}_1'(p,q) = \rho_2(p,q)e^{i\varphi_1(p,q)} \qquad \text{et} \qquad \hat{u}_2'(p,q) = \rho_1(p,q)e^{i\varphi_2(p,q)}.$$

Autrement dit, on passe de u_i à u_i' en conservant les phases issues de u_i mais en utilisant les modules issus de l'autre image. Écrivez une fonction `[U,V]=exchange_phase(u,v)` qui réalise l'échange des phases entre deux images de même taille, et appliquez-là à plusieurs couples d'images.

Quel est l'effet de l'échange du module et des phases ? Parvient-on à reconnaître l'image de départ quand on conserve sa phase ou quand on conserve son module ?

2) Une autre manière de comprendre l'information codée dans la phase de la transformée d'une image est de rendre celle-ci complètement aléatoire (i.i.d. uniforme sur $[0, 2\pi[$), en conservant seulement la contrainte d'antisymétrie ($\varphi(-p,-q) = -\varphi(p,q)$) pour maintenir une image réelle. Cette opération est en fait très utile pour synthétiser une catégorie particulière de microtextures, les textures à phase aléatoire. Pour synthétiser de telles textures, une possibilité est de partir d'une image de texture u, d'extraire sa composante périodique p (fonction perdecomp de la toolbox iptools), puis de générer de nouvelles versions de cette texture en randomisant les phases de p (fonction randphase de la toolbox). Si l'image de départ u n'est pas une texture, on la projette ainsi dans l'espace des textures à phase aléatoire.

Effectuez cette opération sur des images classiques, puis sur des images de texture (trouvées sur le web par exemple). Quelles sont les types de textures les mieux reproduites ? Pouvez-vous expliquer pourquoi ? Que se passe-t-il si l'on travaille directement avec l'image de départ au lieu de sa composante périodique ?

3) Une autre façon de procéder pour créer des textures artificielles est de créer des images binaires très simples (disque noir sur fond blanc, rectangle, damier, etc.) puis de projeter ces images dans l'espace des textures à phase aléatoire. Inspirez-vous de l'example ci-dessous pour produire de telles images et générez ensuite les textures associées. Ce procédé est appelé *Spot noise*.

```
n = 512; I = -n/2:-n/2+n-1;
[X,Y] = meshgrid(I,I);
R = sqrt(X.^2+Y.^2);
u = (R<30);
imshow(u);
imshow(randphase(u),[]);
```

L'image texture.pgm a été obtenue par ce procédé à l'aide d'une ellipse pleine. Estimer, à l'aide de mesures effectuées manuellement sur la transformée de Fourier de l'image, les paramètres de cette ellipse (orientation et longueurs des axes, ou bien équation cartésienne). On pourra commencer par simuler une image du même type, et analyser empiriquement le lien entre les paramètres de l'ellipse de départ et des quantités faciles à mesurer sur la transformée de Fourier de la texture obtenue.
