Exercice 25 (convolutions itérées)

On sait que l'itération d'une convolution par un noyau isotrope positif produit asymptotiquement le même effet que l'équation de la chaleur. En dimension 1, ceci signifie qu'un même noyau convolé plusieurs fois avec lui même a tendance à prendre l'allure de la fonction de Gauss. Nous allons vérifier ce résultat expérimentalement. Créez un signal simple vérifiant les contraintes suivantes :

a) la somme des valeurs est 1;

b) l'ensemble des valeurs est symétrique (palindrome);

c) la dimension est impaire.

Par exemple,

```
s = [3,2,0,1,4,0,4,1,0,2,3]/20;
```

Ensuite, convolez le signal que vous venez de créer $n-1$ fois avec lui-même (utilisez la fonction `t=conv(s,k)` pour calculer $t = s * k$) et visualisez le résultat au moyen de la commande `plot`. Appliquez le procédé pour des valeurs de n de plus en plus grandes et vérifiez que le résultat est conforme à vos attentes.

Essayez la même chose pour plusieurs signaux non positifs (mais qui vérifient les trois conditions a,b,c ci-dessus). Qu'observez-vous ? Pouvez-vous le justifier théoriquement ?

Exercice 26 (équation de la chaleur)

Comme nous l'avons vu en cours, tout filtre linéaire positif isotrope est un schéma numérique pour l'équation de la chaleur

$$\frac{\partial u}{\partial t} = \Delta u.$$

Nous pouvons donc utiliser un schéma très simple à quatre voisins, défini par

$$u^{k+1}(i,j) = u^k(i,j) + s\left(u^k(i+1,j) + u^k(i-1,j) + u^k(i,j+1) + u^k(i,j-1) - 4u^k(i,j)\right)$$

pour k variant de 1 à n (il y a donc n itérations). Ce schéma est réalisé par la fonction `heat` de la toolbox iptools ; essayez par exemple

```
u = double(imread('lena.pgm'))/255;
v = heat(u,20,0.1);
imshow(v);
```

1) En supposant que le pas de discrétisation de l'image est infinitésimal, quelle est la relation entre s, n, et l'échelle t de filtrage théorique par l'équation de la chaleur ?

Vérifiez que l'on obtient visuellement le même résultat lorsque l'on augmente s et diminue n en gardant t constant dans cette relation.

2) Estimez expérimentalement la valeur maximale que l'on peut utiliser pour s, puis donnez-en la valeur théorique. Qu'observe-t-on lorsque la valeur maximale est dépassée ?

Exercice 27 (inversion du temps)

La commande `heat` peut aussi être utilisée avec une valeur de s négative, pour "inverser le temps" dans l'équation de la chaleur. Par exemple, on peut s'attendre à ce que l'image `u2` produite par

```
u1 = heat(u ,20, 0.1);
u2 = heat(u1,20,-0.1);
```

soit très proche de l'image initiale. Essayez de vous faire une idée de la stabilité de ce procédé en

1. augmentant le nombre d'itération n
2. augmentant le pas $|s|$
3. perturbant légèrement (sans impact visuel) l'image `u1`, par exemple :
   - en quantifiant les niveaux de gris (`q = 0.01; u1 = q*floor(heat(u,20,0.1)/q);`)
   - ou en rajoutant un bruit imperceptible à l'œil (`u1 = heat(u,20,0.1)+0.01*randn(size(u));`)

1) Qu'observez-vous (avec cview, utilisez le bouton droit de la souris pour effectuer un "zoom" local) ? Quelles sont les valeurs critiques des paramètres ? Pouvez-vous justifier théoriquement ce manque de stabilité ?

2) Malgré sa sensibilité élevée, ce procédé peut quand même être appliqué pour de petites échelles afin d'obtenir un effet de "déflouage" (*deblurring*). Comparez l'image `u` avec celle obtenue par

```
v = heat(u,10,-0.02);
```

Appliquez ce procédé pour rendre le plus net possible quelques unes des images à votre disposition. Donnez dans chaque cas les valeurs de s et n que vous avez utilisé. Pouvez-vous expliquer pourquoi ce procédé de déflouage peut rester efficace même lorsque le noyau de convolution à inverser n'est pas Gaussien ?
