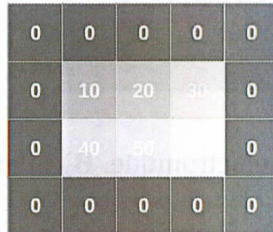


1. Convolution

A



B

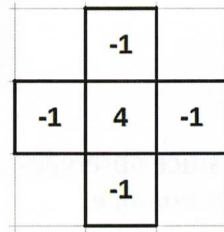


Fig.1 : Convolution d'une image. A) Image originale en niveaux de gris de dimension 5x4 pixels. B) Masque de convolution en croix et ses coefficients.

- **Question 1.1** : Calculez la convolution de l'image Fig1A avec le masque (Fig.1B). Seuls les pixels différents de zéro doivent être calculés. Indiquez les calculs intermédiaires. Il s'agit de calculer **à la main** les valeurs et non pas de proposer un algorithme.

2. Bruit de fond

On se propose de générer une image de bruit de fond ayant une distribution Gamma (ou Erlang). Cette distribution est calculée selon la formule en Fig. 2A à partir des paramètres k et λ .

- **Question 2.1** : Implanter en Javascript la fonction **gamma(imp, k, lambda)** qui prend trois arguments : Une *ImagePlus*, les constantes k et λ et modifie l'image par un bruit de fond à distribution gamma. Par exemple, le code suivant génère une image dont l'histogramme est en Fig. 2B.

```
let imp = IJ.createImage("noise", "32-bit black", 256, 256, 1);
gamma(imp, 2, 2) ; // Fill the imp with pixels according to formula Fig.2A
imp.show() ;
```

Note : On utilisera `Math.log(v)` pour le logarithme et `Math.random()` pour générer un nombre aléatoire entre 0 et 1.0.

$$E(k, \lambda) \approx -\frac{1}{\lambda} \ln \prod_{i=1}^k U_i$$

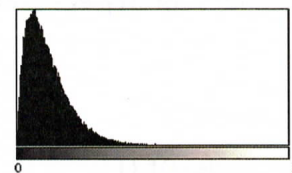
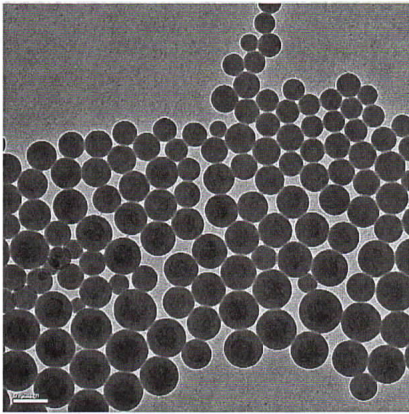


Fig.2 : Bruit de fond à distribution Gamma (Erlang). A) Formule du calcul d'un pixel de l'image. U correspond à une valeur aléatoire entre 0 et 1. B) Histogramme d'une image calculée avec les paramètres $k = 2$, $\lambda = 2.0$

3. Traitement et Analyse d'images

On se propose de traiter et d'analyser des images de microscopie électronique comme celle présentée dans la Fig. 1 avec le logiciel ImageJ.

A



B

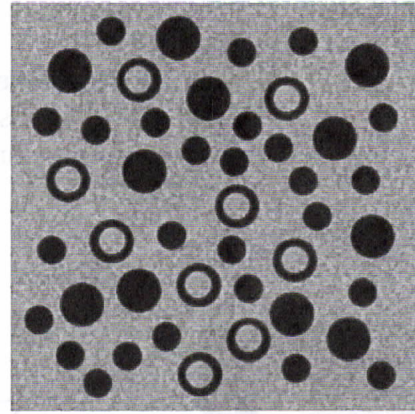


Fig.3 : A) Image originale de particules de silice observées au microscope électronique. B) Image Test simulant une image de microscopie électronique.

- **Question 3.1 :** On ne s'intéresse qu'à la **Fig. 3A** pour cette question. Comptage des particules. Indiquez un protocole (i) de traitement puis (ii) d'analyse pour obtenir les coordonnées X,Y des centres ainsi que les rayons des particules dans l'image de la Fig.1A **SANS UTILISER** « *Analyze Particles* » . Pour chacune des étapes, décrire succinctement quel est le **principe** de la fonction, utilisée et **justifiez** son utilisation par rapport aux défauts potentiels de l'image originale.
- **Question 3.2. :** On ne s'intéresse qu'à la **Fig. 3B** pour cette question. Ce projet contient plusieurs centaines d'images du même type, on souhaiterait classer ces particules par une méthode de *Machine Learning* (ML). En vous basant sur ce que vous avez vu en cours et dans le livre **The Master Algorithm**, on se propose d'utiliser une méthode **non supervisée** pour trier ces particules à partir de mesures réalisées avec ImageJ (Commande *Analyze Particles*).
 - Quelles sont les mesures (*Measurements*) les plus adaptées pour trier vos particules? Indiquez leur principe. Justifiez votre choix.
 - Choisir un algorithme de ML, présentez succinctement son principe et justifiez votre choix.
 - A quoi pourrait servir l'Analyse en Composantes Principales (ACP ou PCA en anglais) dans cette étude ?

4. Annexes

Propriétés et méthodes **autorisées** pour les questions de programmation en JavaScript.

Array Propriétés array.length Méthodes Array.indexOf() Array.push(element) Array.reverse() Array.slice(start,end)	Math Propriétés Math.PI Méthodes Math.log(value) Math.random() Math.tan(value) Math.floor(value)
ImagePlus getWidth() getHeight() getProcessor() show()	ImageProcessor getWidth() getHeight() putPixel(x, y, value) getf(index) setf(index, value)