1- Dégradations dans une image :

Lors de l'acquisition, de la transmission ou de la compression d'une image, il peut apparaître de nombreuses dégradations telles que le bruit (additif ou impulsionnel).

Pour ajouter du bruit à une image sous MATLAB, on utilise la commande **imnoise**

Exercice

A. Ouvrir une image, la stocker dans une matrice img.
B. Appliquer sur l'image img un bruit blanc Gaussien et stocker le résultat dans une matrice Y1.
C. Appliquer sur la même image un bruit "poivre et sel" et stocker le résultat dans une matrice Y2.

```
% img: image renormalisee (valeur entre 0 et 1)
»Y1 = imnoise(img, 'gaussian')
»Y2 = imnoise(img, 'salt & pepper')
```

D. Afficher sur une même figure les images img, Y1, Y2.
E. Tracer sur la même figure les signaux des images img, Y1, Y2.

```
% methode 1
» figure
»plot(1:N,x1)
»hold on
»plot(1:N,x2,'g')
»plot(1:N,x3,'r')
% methode 2
»figure
»plot(1:N,x1,1:N,x2,1:N,x3)
```

2- Filtrage d'une image

On distingue deux types de filtrage : Linéaire et non Linéaire.

Pour réaliser un filtrage linéaire, il faut d'abord définir le masque h à utilisé. Pour cela, soit on le définit de façon analytique, soit on utilise la fonction "fspecial" de MATLAB

```
% Methodes analytiques
» h = ones(3,3)/9;    % Filtre moyenneur de taille 3 x 3

% Methodes en utilisant fspecial
» h = fspecial('average',[3 3]);       % Filtre moyenneur de taille 3 x 3
```

On réalise ensuite le filtrage de l'image grâce à la commande imfilter: `» Y = imfilter(Y1, h, 'replicate');`

La syntaxe de la fonction "imfilter" est ; B = imfilter(A, h) ou B = imfilter(A, h, options,...)

Le filtrage non-linéaire que nous allons considérer ici est le filtrage médian, qui peut être réalisé grâce à la commande "medfilt2" :

```
» X = medfilt2(Y1, [3 3]);
```

3- Métrique d'évaluation :

Afin de pouvoir quantifier la qualité du débruitage, on va utiliser une mesure objective appelée Peak Signal to Noise Ratio (PSNR) et définie par :

$$PSNR = 10\log_{10}\left(\frac{R^2}{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[x^o(m,n) - x^d(m,n)\right]^2}\right)$$

Où x^o et x^d sont respectivement les images originelles et débruitées et R la dynamique du signal (valeur maximale possible pour un pixel).

Si l'on travaille sur des images renormalisées, on a R = 1 et le PSNR peut se calculer facilement grâce à l'instruction :

```
% Y1 : image originelle, Y : image debruitee
» PSNR= -10*log10(std2(Y1 - Y));
```

Calculer le PSNR pour les deux simulations précédemment réalisées.