

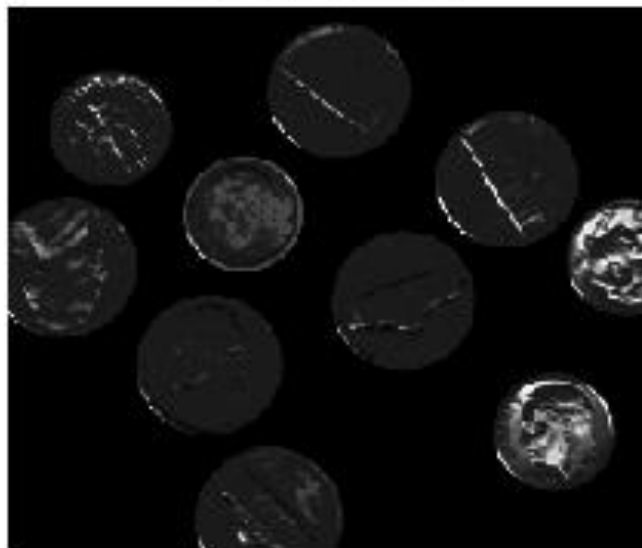
TP2 Traitement et synthèse d'image :  
Morphologie mathématique

# 1 Transformations d'histogramme

En segmentant l'image d'origine suivante :

```
clear variables;  
close all;  
  
%question 1  
Im = imread ('pieces.png');  
[h,w]=size(Im);  
  
Im2 = im2double(Im);  
  
figure(1);  
imshow(Im, []);  
title 'Image origine'
```

**Image origine**



```
%question2  
m1 = 50;  
m2 = 180;  
M1 = 0;  
M2 = 0;  
eps = 1;  
while ((abs(M1-m1)>eps) || (abs(M2-m2)>eps))  
    Labels = zeros(h,w);  
    M1 = m1;
```

```

M2 = m2;
L1 = 0;
L2 = 0;
P1 = 0;
P2 = 0;
for i=1:h;
for j =1:w;
if (abs(M1 - 255*Im2(i,j)) < abs(M2-255*Im2(i,j)));
    Labels(i,j) = 1;
    L1 = L1 + Im2(i,j)*255;
    P1 = P1 + 1;

else
    Labels(i,j) = 2;
    L2 = L2 + Im2(i,j)*255;
    P2 = P2 +1;

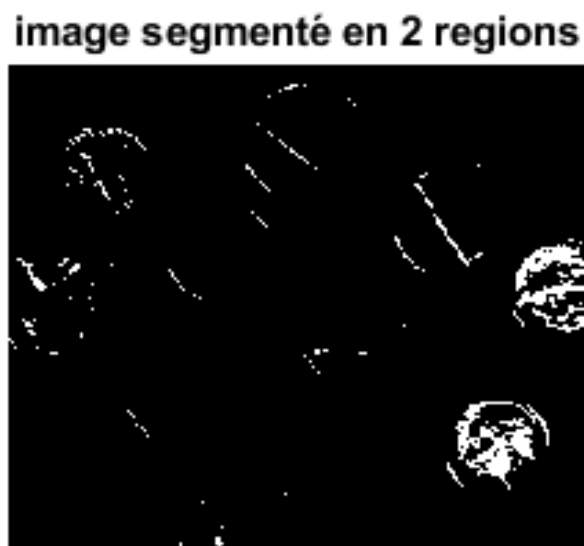
end
end
end
m1 = L1/P1;
m2 = L2/P2;
M1 = m1;
M2 = m2;

end

Labels = Labels-1;
figure(2);
imshow(Labels, []);
title 'image segmenté en 2 regions'

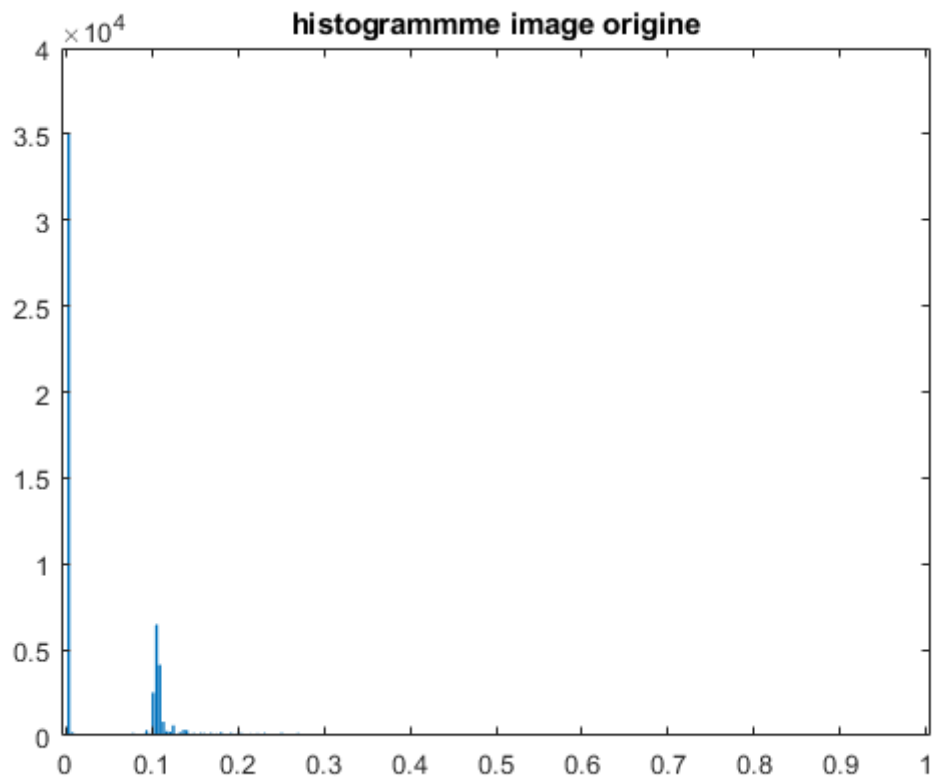
```

On obtient l'image suivante :



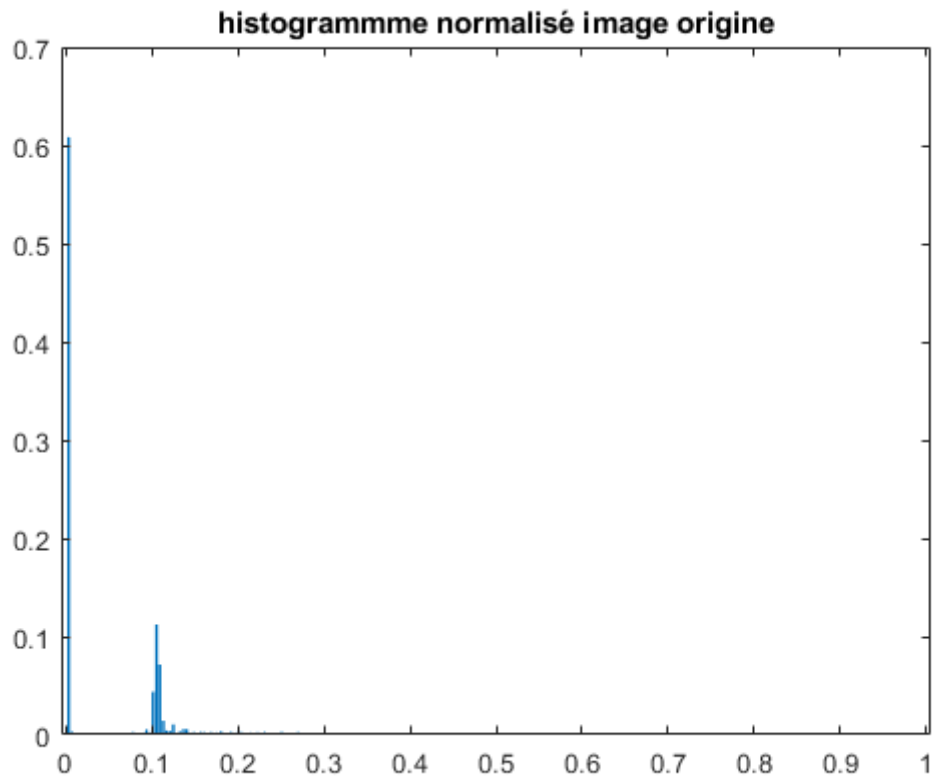
On remarque que l'image obtenue n'est pas satisfaisante car d'après l'histogramme ci-dessous on remarque que les pixels de la photo sont principalement noirs ou sombres.

```
%question 3
[HistIm,y]=imhist (Im2);
figure(3);
bar(y,HistIm)
title 'histogramme image origine'
```



Pour plus de lisibilité on peut normaliser l'histogramme en divisant par le nombre pixels :

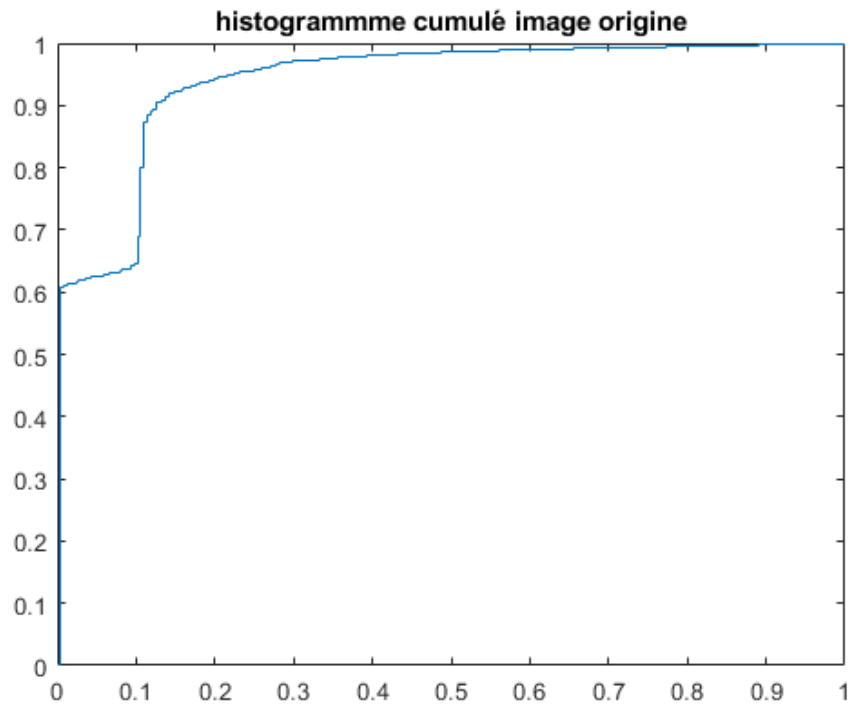
```
%question 4
HistNorm=HistIm/ (h*w);
figure(4);
bar(y,HistNorm)
title 'histogramme normalisé image origine'
```



On peut aussi remarquer à l'aide de cet histogramme que si on prend les deux points d'origine aux alentours de 2 et 30 ( $\approx 0.1 \cdot 255$ ) on obtient un résultat un peu plus cohérent.

On calcule ensuite l'histogramme cumulé qui nous servira ensuite pour la reconstruction de l'image par égalisation.

```
%question 5
HistCum=zeros(1,256);
HistCum(1)=HistNorm(1);
for i = 2:256
    HistCum(i)=HistNorm(i)+HistCum(i-1);
end
figure(5);
stairs(y,HistCum)
title 'histogramme cumulé image origine'
```



On remarque bien que la majorité des pixels sont noir ou foncé.

Ensuite à l'aide de l'histogramme cumulé on modifie l'image d'origine pour avoir des pixels plus équilibrés sur l'échelle [0,255].

```
%question 6
ImEga=HistCum (Im2.*255+1);
figure(6)
imshow(ImEga,[])
title 'image égalisée'
```

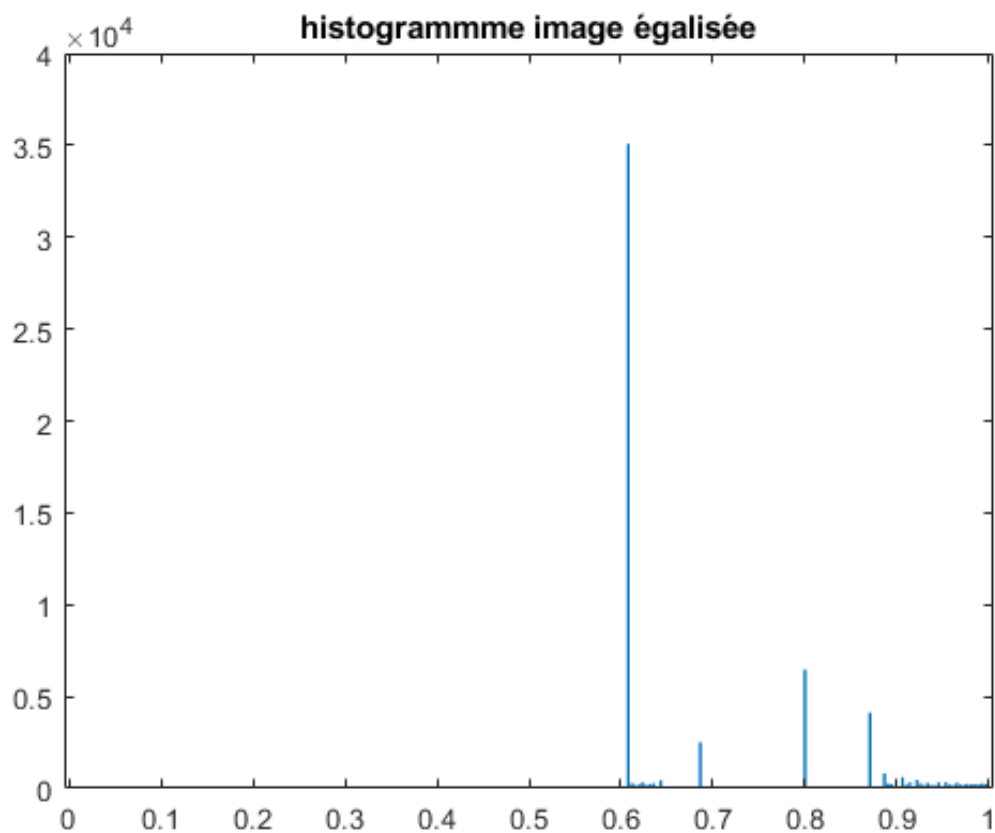
Nous obtenons finalement l'image suivante :



On a donc une image visuellement beaucoup moins sombre que l'image d'origine.

On calcul ensuite l'histogramme de la nouvelle image obtenue.

```
%question 7
[HistEga,x]=imhist(ImEga);
figure(7)
bar(x,HistEga)
title 'histogramme image égalisée'
```



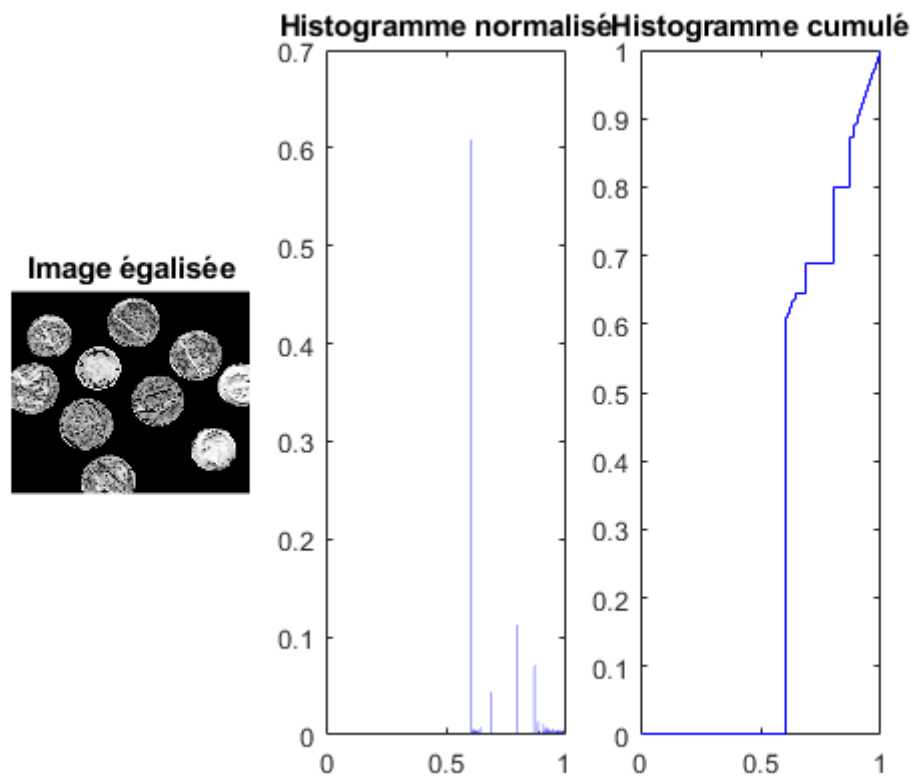
On observe donc que les pixels sont plus étalés que dans l'image d'origine ce qui explique que l'image obtenue est plus claire.

```

%question 8
HistNormEga=HistEga/(h*w);

figure(8);
HistCumEga=zeros(1,256);
HistCumEga(1)=HistNormEga(1);
for i = 2:256
    HistCumEga(i)=HistNormEga(i)+HistCumEga(i-1);
end
subplot(131)
imshow(ImEga,[])
title('Image égalisée')
subplot(132)
bar(x,HistNormEga,'b')
title('Histogramme normalisé')
subplot(133)
stairs(x,HistCumEga,'b')
title('Histogramme cumulé')

```

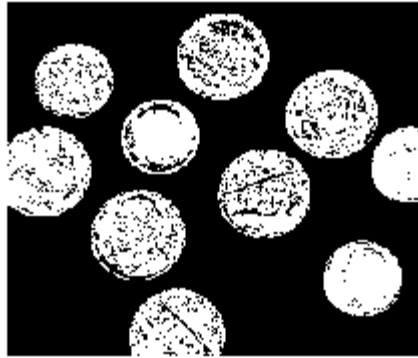


Qu'est-ce que c'est beau !! <3

On réalise ensuite la méthode des K-means sur l'image normalisée ce qui nous permet d'obtenir l'image ci-dessous :



image égalisée segmentée en 2 regions



On obtient donc bien une jolie segmentation des pièces d'origine ce qui est bien ce que l'on voulait. Donc la méthode d'égalisation d'histogramme se révèle comme une bonne solution pour des images trop sombre (ou potentiellement trop claire).

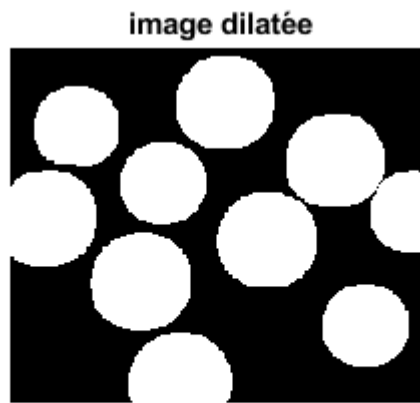
Néanmoins les motifs sur les pièces ne sont pas encore très propres, et elles ne sont pas vraiment rondes comme on s'attendait à ce qu'elles soient.

## 2 Morphologie mathématique : granulométrie

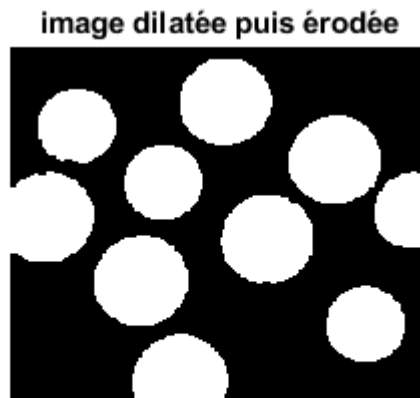
On cherche ensuite à réaliser la courbe de granulométrie des pièces de l'image d'origine, pour cela nous allons utiliser l'image segmentée obtenue à la fin de la partie précédente.

L'image n'étant pas tout à fait homogène nous allons « boucher » les trous à l'aide d'une dilatation, ensuite pour revenir à la bonne taille de pièce on érode l'image obtenue.

```
%question 1
SE= strel('disk',3);
SE2= strel('disk',2);
ImDilate= imdilate(Labels2,SE);
ImErode=imerode(ImDilate,SE);
figure(10);
imshow(ImDilate)
title 'image dilatée'
```



```
figure(11);
imshow(ImErode)
title 'image dilatée puis érodée'
```



On vérifie bien que les cercles obtenus correspondent bien aux cercles de l'image d'origine et sont bien homogènes.

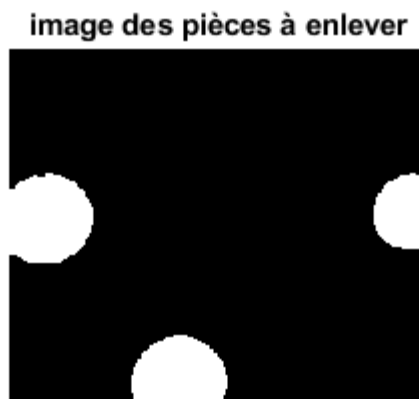
Notre but étant de déterminer la taille des pièces on veut supprimer les pièces coupées sur la photo car on risque de mal estimer leur taille.

Pour cela on utilise un marqueur qui fait le contour de l'image qui va pouvoir détecter les pièces coupées.

```
%question 2
Marqu=zeros(h,w);
Marqu(1,:)=1;
Marqu(:,1)=1;
Marqu(h,:)=1;
Marqu(:,w)=1;
```

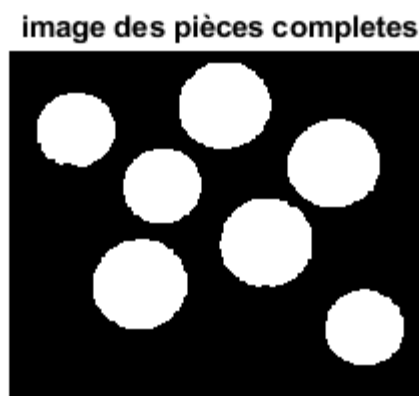
```
%question 3
Pieces_a_enlever=Marqu.*ImErode;
ImRel=imreconstruct (Pieces_a_enlever,ImErode);
figure(12)
imshow(ImRel)
title 'image des pièces à enlever'
```

En réalisant la reconstruction on obtient :



On soustrait donc ensuite les pièces à enlever à l'image précédente pour obtenir :

```
figure(13)
Tuveuxpascreerunnouvelleimage =ImErode-ImRel;
imshow(Tuveuxpascreerunnouvelleimage)
title 'image des pièces completes'
```

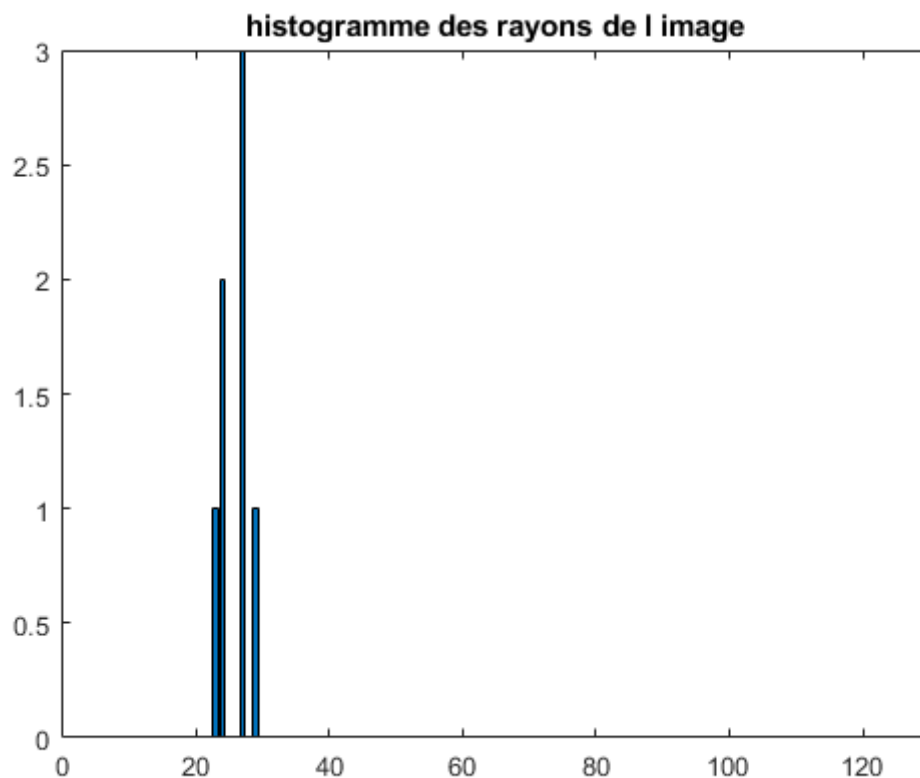


Ensuite nous pouvons utiliser `bweuler` afin de compter le nombre de pièce qu'il y a dans l'image. Cette fonction va nous servir à déterminer le rayon des pièces : en érodant les pièces d'un certain rayon  $R$  que l'on va faire augmenter, des pièces vont finir par disparaître ce qui nous permettra de

connaître leur rayon, puisqu'il s'agit de  $R$  le rayon utilisé pour l'érosion. On arrête de faire augmenter  $R$  lorsque la fonction `bweuler` ressort 0, c'est-à-dire lorsqu'il n'y a plus de pièces sur le l'image.

```
%question 4
bweuler(Tuveuxpascreerunenouvelleimage)

SE2 = [0,1,0;1,1,1;0,1,0];
rayon=1;
nb_pieces = zeros(1, floor(max(h, w)/2));
i = 1;
while ~(isequal(Tuveuxpascreerunenouvelleimage, zeros(h,w)))
    nb_pieces(i) = bweuler(Tuveuxpascreerunenouvelleimage);
    Tuveuxpascreerunenouvelleimage
=imerode(Tuveuxpascreerunenouvelleimage,SE2);
    rayon=rayon+1;
    i = i+1;
end
figure;
tailles = nb_pieces(1:end-1)-nb_pieces(2:end);
bar(1:length(tailles), tailles);
title 'histogramme des rayons de l image'
```



On remarque donc qu'il y a 1 pièce de rayon 23 pixels, 2 pièces de rayons 24, trois pièces de rayons 27 et 1 pièce de rayons 29. On obtient donc bien au total 7 pièces et le nombre le nombre de taille identique semble cohérente à l'image d'origine.