

Matériel et méthodes : programme sur Matlab

```
clear all;  
clc;
```

```
% chargement des images
```

```
liste_noms=rdir('*.*tiff');  
dY0 = double(imread(liste_noms{1}));  
dY = double(imread(liste_noms{2}));  
[nx,ny] = size(dY0); % Initialisation de nx (dimension verticale) et ny (dimension horizontale)
```



```
depl_max = 5;
```

```
% échantillonnage en petites zones
```

```
l_ech_x = 18; %longueur selon x de l'échantillonnage (hauteur des petites zones)  
l_ech_y = 18; %longueur selon y de l'échantillonnage (largeur des petites zones)  
milieu_x=floor((l_ech_x+l_ech_x+(2*depl_max)-1)/2)+1; %milieu de l'image de corrélation  
milieu_y=floor((l_ech_y+l_ech_y+(2*depl_max)-1)/2)+1;  
nb_ech_x = floor((nx-(2*depl_max))/l_ech_x); %nombre d'échantillon selon x  
nb_ech_y = floor((ny-(2*depl_max))/l_ech_y); %nombre d'échantillon selon y  
decallage_x = mod((nx-(2*depl_max)),l_ech_x); %nb de pixels pas compté dans l'échantillonnage  
decallage_y = mod((ny-(2*depl_max)),l_ech_y);
```

```
% itération sur l'ensemble des pas de récurrence choisis
```

```
position = 0; %compteur pour incrémenter le tableau de cellule « resultats »  
for n = [3,4,5,6,7,8]  
    position = position + 1;  
    nb_images = 49-n; %exclusion des pasème dernière images pour ne pas être hors indice
```

```
%initialisation des matrices contenant les déplacements de chaque petite zone
```

```
pos_max_x=zeros(nb_ech_x,nb_ech_y); % déplacements en pixels  
pos_max_y=zeros(nb_ech_x,nb_ech_y);  
deplacement_x=zeros(nb_ech_x,nb_ech_y); %déplacements en mm  
deplacement_y=zeros(nb_ech_x,nb_ech_y);
```

```
%initialisation du tableaux contenant la profondeur du creux à chaque itération
```

```
profondeur_eq= zeros(1,floor(nb_images/pas));  
eq=0; %compteur pour incrémenter le tableau « profondeur_eq »
```

```
% itération sur l'ensemble des images de la série (les images sont prises deux par deux)
```

```
for i = 1:pas:nb_images  
    eq=eq+1;  
    dY0 = double(imread(liste_noms{i}));  
    dY = double(imread(liste_noms{i+pas}));
```

```
% itération sur chaque petite zone de l'image définie lors de l'échantillonnage
```

```
xi=0; %compteurs pour incrémenter les matrices de déplacements  
yj=0;  
for i = l_ech_x+decallage_x+depl_max+1:l_ech_x:nx-depl_max  
    xi=xi+1;
```

```

yj=0;
for j = l_ech_y+decallage_y+depl_max+1:l_ech_y:ny-depl_max
    yj=yj+1;

%incrémentation des petites zone i équivalentes dans les 2 images et valeur moyenne mise à 0
    mini_dY0 = dY0(i-l_ech_x+1-depl_max:i+depl_max,j-
l_ech_y+1-depl_max:j+depl_max);
    mini_dY0 = mini_dY0 - mean(mean(mini_dY0));
    mini_dY = dY(i-l_ech_x+1:i,j-l_ech_y+1:j);
    mini_dY = mini_dY - mean(mean(mini_dY));

%produit de corrélation
    corr = xcorr2(mini_dY0,mini_dY);

%recherche des coordonnées du maximum de corrélation
    [maximum_1,max_corr_y_1]= max(max(abs(corr),[],1));
    [maximum_1,max_corr_x_1]= max(max(abs(corr),[],2));

%mise à 0 du pic autour du premier maximum de corrélation
    [nx_corr,ny_corr] = size(corr);
    large = 4; %introduction d'un bord de 0 pour éviter d'être hors indice lors de la mise à 0
    corr_2 = zeros(nx_corr+(2*large),ny_corr+(2*large));
    corr_2(large+1:nx_corr+large,large+1:ny_corr+large) = corr;
    corr_2(max_corr_x_1:max_corr_x_1+(2*large),max_corr_y_1:max_corr_y_1+(2*large))
= 0;

%recherche du 2ème maximum
    [maximum_2,max_corr_y_2]= max(max(abs(corr_2),[],1));
    [maximum_2,max_corr_x_2]= max(max(abs(corr_2),[],2));
    max_corr_y_2 = max_corr_y_2-large;
    max_corr_x_2 = max_corr_x_2-large;

%calcul des distances entre l'origine de l'image de corrélation et les maxima de corrélation
    distance_1 = ((milieu_x-max_corr_x_1)^2) + ((milieu_y-max_corr_y_1)^2);
    distance_2 = ((milieu_x-max_corr_x_2)^2) + ((milieu_y-max_corr_y_2)^2);

%choix du bon maximum de corrélation
    bis = (5/6)*maximum_1; %intensité suffisamment élevée
    if distance_2 < distance_1 & maximum_2 > bis %distance la plus courte
        max_corr_x = max_corr_x_2;
        max_corr_y = max_corr_y_2;
    else
        max_corr_x = max_corr_x_1;
        max_corr_y = max_corr_y_1;
    end

%incrémentation des matrices de déplacement avec les coordonnées du point choisi
    pos_max_x(xi,yj)=pos_max_x(xi,yj) + milieu_x-max_corr_x;
    pos_max_y(xi,yj)=pos_max_y(xi,yj) + milieu_y-max_corr_y;
    déplacement_x(xi,yj)=déplacement_x(xi,yj) + (milieu_x-max_corr_x)*(29.0/480.0);
    déplacement_y(xi,yj)=déplacement_y(xi,yj) + (milieu_y-max_corr_y)*(38.0/640.0);
end

```

```
end
```

```
%valeur moyenne mise à 0 (détermine l'origine de l'axe des z de la surface)
```

```
deplacement_x=deplacement_x-mean(mean(deplacement_x));  
deplacement_y=deplacement_y-mean(mean(deplacement_y));
```

```
% calcul de la hauteur de la surface avec la fonction surfheight
```

```
dr.x = [1:nb_ech_x]*l_ech_x*0.06;  
dr.y = [1:nb_ech_y]*l_ech_y*0.06;  
dr.vx=deplacement_x;  
dr.vy=deplacement_y;  
dr.unitx='mm';  
dr.unity='mm';  
dr.namex='x';  
dr.namey='y';  
dr.unitvx='m/s';  
dr.unitvy='m/s';  
dr.namevx='u_x';  
dr.namevy='u_y';  
dr.name='toto';  
dr.setname='tata';  
dr.history={['loadvec('' dr.namex '')']};
```

```
g = surfheight(dr,196.2,183.1,1.4,'submean');  
g.w = g.w - mean(g.w(:));
```

```
%filtrage du résultat pour lisser la surface (éliminer le bruit d'intégration)
```

```
ff= bwfilterf(g,7,6) ;
```

```
% tracé en 3D de la surface
```

```
% surf(ff.y,ff.x,ff.w)  
% shading interp  
% colorbar  
% title('h (mm)')  
% xlabel('x (mm)')  
% ylabel('y (mm)')  
% axis([0 40 0 30 -0.1 0.1])  
%  
% pause(0.01)
```

```
% calcul de la profondeur du creux
```

```
profondeur = max(max(ff.w))-min(min(ff.w));  
profondeur_eq(eq) = min(min(ff.w));
```

```
end
```

```
% tracé de l'évolution de la profondeur du creux en fonction du nombre d'itérations
```

```
figure,plot(profondeur_eq)
```

```
title('évolution de la profondeur du creux')  
xlabel('nombre d'itérations')  
ylabel('profondeur (mm)')
```

```
%incrémentation du tableau des profondeurs calculé dans le tableau de cellule « resultats »  
resultats{position} = profondeur_eq;
```

```
end
```

```
%enregistrement du tableau de cellule  
save('resultats','resultats');
```