

```

%% PSE – Resistance de vagues – Traitement numerique
clc;
clear all;

%% Recuperer les donnees sous image J: Moyenner les pixels

% Entrée: fichier .txt du Reslice sous Image J.
% x(:,1) = pixel de la position de la goutte
% x(:,2) = numéro de l'image

%longueurfichier = nb de positions/images différentes de la goutte pris en
%compte dans le reslice pour l'algorithme.

longueurfichier=50; %Attention: doit etre plus petit que posmoy, adapter si besoin.
f=rdir('*.*txt');
positionmoyennee=zeros(longueurfichier,length(f));
vitesse=zeros(longueurfichier-1,length(f));
acceleration=zeros(longueurfichier-1,length(f));

for l=1:length(f)
    x=load(f{l});
    N=length(x(:,1));
    pos=x(:,1);
    image_initiale=x(1,2)
    t=x(:,2)-x(1,2); %correspond au numero de l'image
    M=max(t)+1;%nombre d'images différentes (l'indexation commence à 0)
    a=zeros(M,1); % vecteur qui compte la frequence d apparition d une image
    a=a+1; % on a au moins une position relevee par image

    pos_somme=zeros(M,1);
    pos_moy=zeros(M,1);
    var=zeros(M,1);
    ecart_type=zeros(M,1);
    j=1;

    % si on relève deux positions sur la même image, on les somme pour en faire
    % la moyenne.

    for i=1:N-1
        if t(i)==t(i+1);
            a(j)=a(j)+1;
            pos_somme(j)= pos_somme(j)+pos(i);
        else
            pos_somme(j)= pos_somme(j)+pos(i);
            j=j+1;
        end
    end

    for k=1:M
        pos_moy(k)=pos_somme(k)/a(k);
    end

    u=1;
    for i=1:N-1
        if t(i)==t(i+1);
            a(u)=a(u)+1;
            var(u)= var(u)+(pos(i)-pos_moy(t(i)+1))^2;
        else
            var(u)= var(u)+(pos(i)-pos_moy(t(i)+1))^2;
            ecart_type(u)=sqrt(var(u)/a(u));
            u=u+1;
        end
    end
end

```

```

end
%Ici, on a moyenné les positions : on plus qu'une seule position de la goutte
% par image.

taille1=length(pos_moy)-(longueurfichier-1);
taille2=length(pos_moy);
positionmoyennee(:,l)=pos_moy(taille1:taille2);

%% Affichage :

P=longueurfichier; %paramètre qui permet de régler la partie à afficher.

% échelle (bout métallique qui dépasse de l'aiguille): 1.3 cm correspondant
% à 69 pixels --> on fabrique la vitesse en cm/sec.
% nombre d'images par seconde = 105
positionrenverse=smooth(pos_moy);
position = positionrenverse(end:-1:1);
%tout comme les positions qui sont lues par la fin, il faut inverser les écarts
%types.
erreur = ecart_type(end:-1:1);
variance=var(end:-1:1); % va servir pour calculer les incertitudes de vitesse.
position_final=position(1:P)-position(1,1);
distance=position_final*1.3/69;
v=(diff(position))*1.3/69*105;
acc=(diff(smooth(v)))*105;
err=erreur(1:P)*1.3/69; %erreur sur la position%

vitesse(:,l)=v(1:P-1); %avoir les bonnes dimensions pour la suite.
acceleration(:,l)=acc(1:P-1);
v_final2=v(1:P-1);
end

% affichage de la position d'une goutte + incertitude de la position.
temps1=[0:1:length(position_final)-1]*0.01;
figure (1)
subplot(3,1,1)
errorbar(temps1,distance,err)
hold on
title('Position of the drop as a function of time )
xlabel('Time (s)')
ylabel('Distance to initial point (cm)')

%Calcul de la vitesse moyenne et accélération moyenne + barres d'erreurs
%(pour plusieurs gouttes à la fois)

indmax=zeros(1,length(f));

for l=1:length(f)
    [Maximum,Indice]=max(vitesse(:,l));
    [maximum,indice]=max(acceleration(:,l));
    indmax(l)=Indice;
end

Maxindicevitesse=max(indmax);
Minindicevitesse=min(indmax);
Deltavitesse=(Maxindicevitesse-Minindicevitesse);

vmoy=zeros(P-1-Deltavitesse,1);
accmoy=zeros(P-1-Deltavitesse,1);

temps2vitesse=[0:1:P-Deltavitesse-2]*0.01+0.06;

```

```

errvitesse=zeros(P-1-Deltavitesse,1);

temps2acc=[0:1:P-Deltavitesse-2]*0.01+0.06;
erracc=zeros(P-1-Deltavitesse,1);

for l=1:length(f)
    vmoy=vmoy+vitesse(indmax(l)-(Minindicevitesse-1):(P-1)-Maxindicevitesse+indmax(l),
    l);
    accmoy=accmoy+acceleration(indmax(l)-(Minindicevitesse-1):(P-1)-
    Maxindicevitesse+indmax(l),l);
end
vmoy=vmoy/length(f);
accmoy=accmoy/length(f);

for l=1:length(f)
    errvitesse=(vitesse(indmax(l)-(Minindicevitesse-1):(P-1)-Maxindicevitesse+indmax
    (l),l)-vmoy).^2;
    erracc=(acceleration(indmax(l)-(Minindicevitesse-1):(P-1)-Maxindicevitesse+indmax
    (l),l)-accmoy).^2
end
errvitesse=sqrt(errvitesse);
erracc=sqrt(erracc)

subplot(3,1,2)
errorbar(temps2vitesse,smooth(vmoy),errvitesse)
title('Velocity of the drop as a function of time ')
xlabel('Time (s)')
ylabel('Velocity (cm/s)')
subplot(3,1,3)
errorbar(temps2acc,smooth(accmoy),erracc)
title('Acceleration of the drop as a function of time')
xlabel('Time (s)')
ylabel('Acceleration(cm/s^2)')

% Affichage : décélération en fonction de la vitesse.
figure (2)
plot(smooth(vmoy), smooth(-accmoy))
title('Deceleration of the drop depending on velocity ')
xlabel('Velocity (cm/s)')
ylabel('Deceleration(cm/s^2)')
hold on

```