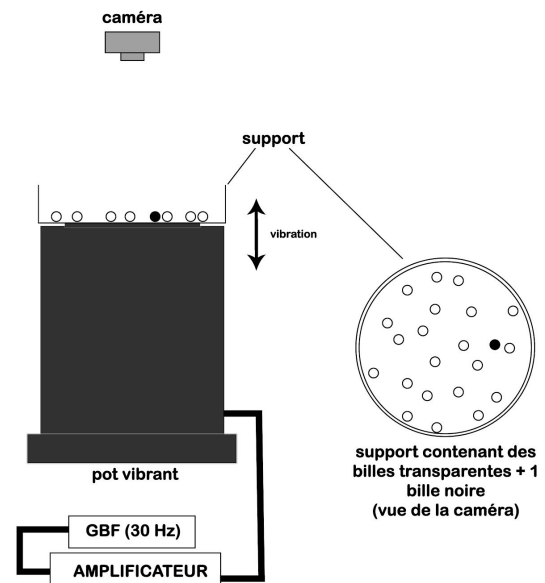


PSE : Longueur moyenne de diffusion dans un mouvement brownien

Cécile Davrinche, Charlotte Durand, Mélanie Ruelle
ESPCI 136ème promotion
15 Mai 2019

1. But de la manipulation

On cherche à caractériser le mouvement brownien, ainsi que les propriétés d'invariance de la distance moyenne de diffusion. Pour cela, on suit le mouvement d'une bille noire soumise à un mouvement brownien au sein d'un cristallin rempli de billes transparentes.



2. Montage

On réalise le montage ci-dessous :

On dispose de :

- Support métallique 40cm x 40cm x 70cm
- Caméra noir et blanc acA1300-200um avec un objectif de focale fixe 1" 16mm/F1.4
- Amplificateur
- Générateur à basse fréquence
- Pot vibrant
- Support circulaire pour le pot vibrant de diamètre 11cm

- Papier adhésif blanc mat
- Tissu noir disposé autour du pot vibrant
- Billes transparentes de diamètre 1mm
- Une bille noire de diamètre 1mm

Les images sont prises de telle sorte à ne voir que le support contenant les billes au-dessus du montage, comme ci-dessous :

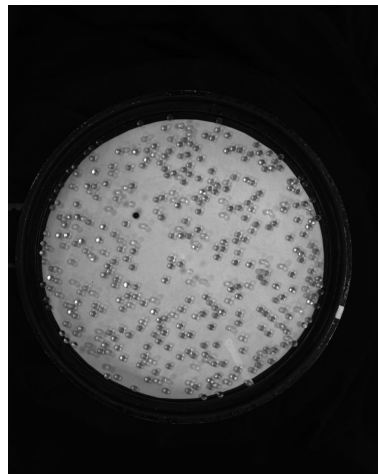


Image prise avec 600 billes dans la cuve

3. Protocole

3.1 Préparation des billes et du support

On peint à la bombe quelques billes en noir. On vérifie bien qu'elles conservent leur forme sphérique. On colle un adhésif blanc mat au fond du support afin de mieux pouvoir distinguer la bille noire.

3.2 Enregistrement des vidéos

On actionne le pot vibrant avec un GBF (30 Hz) relié à un amplificateur (1,8 A, 0,8 V). On règle la caméra de type acA1300-200um avec un objectif de focale fixe 1" 16mm/F1.4 et une fréquence d'acquisition de 8 images par seconde. On enregistre alors plusieurs vidéos d'une trentaine de minutes en faisant varier le nombre de billes dans le support. Nous prenons 16000 images par acquisition.

4. Traitement des données

On analyse ensuite les données sous Matlab avec 2 buts :

- Vérifier que l'on dispose bien d'un mouvement brownien

- Obtenir la distance moyenne de diffusion sur différentes zones du support

Explication du code sous Matlab :

Dans un premier temps, il va falloir suivre la bille noire pour récupérer ses coordonnées sur les différentes images.

Le code ouvre les images, les convertit en noir et blanc, puis grâce à un seuil réglable effectue une conversion binaire de chaque pixel soit en blanc (0) soit en noir (1). Ensuite, il va venir repérer les zones noires de chaque image et enregistrer les coordonnées de son centre dans la matrice `center`.

Dans le cas où le programme n'arrive plus à repérer la bille noire, on rentre (0,0) comme coordonnées du centre. La deuxième section du code permet ensuite de pointer manuellement la bille pour ces images avec la commande `getpts`, en repérant les (0,0) dans la matrice `center`. Puis on sélectionne la zone de l'espace sur laquelle on veut obtenir la distance de diffusion moyenne.

Première partie du programme :

```
%Recuperer les images
rep = '/Volumes/My Passport/200b_08_03_19'; %Dossier dans lequel on
trouve les images
ext = '*.tiff';
chemin = fullfile(rep,ext);
list = dir(chemin);
for i=1:16000
    a=imread(fullfile(rep, list(i).name), ext(3:end));
    centers=imfindcircles(255-a,[6,30],'EdgeThreshold',0.2);
    if isempty(centers)==1
        centers=[0 0]
    end
    center(i,:)=centers(1,:);
    i
end

%% Pointage manuel en complement
coor=center;
rep = '/Volumes/My Passport/200b_08_03_19'; %Dossier dans lequel on
trouve les images
ext = '*.tiff'; %Normalement ça change pas
chemin = fullfile(rep,ext);
list = dir(chemin);
[X,A]=find(coor(:,1)==0);
N=size(A);
N1=N(1);
for i=1:N1
    a=imread(fullfile(rep, list(X(i)).name), ext(3:end));
    imshow(a)
```

```

    [x,y] = getpts
    coor(X(i),:)= [x,y];
end

```

Vérification d'une caractéristique du mouvement brownien :

Soit une durée de diffusion Δt pendant laquelle la bille se déplace à l'intérieur d'une zone définie. Soit d la distance parcourue pendant Δt . Pour des petites durées de diffusion, la bille ne subit pas d'impact avec les autres billes et suit un comportement balistique, tel que d est proportionnel à Δt . Pour des grandes durées de diffusion, la trajectoire de la bille est déterminée par les chocs avec les autres billes : on a alors un mouvement brownien tel que d est proportionnel à \sqrt{t} (comportement diffusif).

On trace alors la droite $\ln(d)$ en fonction de Δt . Si on a un mouvement brownien, on devrait alors voir la pente de la droite diminuer d'un facteur 2 lorsque la durée de diffusion devient assez grande pour qu'il y ait des chocs entre billes.

Dans le programme ci-dessous, d est la variable `pas` et on calcule la distance moyenne parcourue par la bille noire entre deux images séparées de `pas` images, et ce pour `pas` allant de 1 à 16000 (le nombre maximum d'images). Cependant, il est à noter que lorsque le `pas` devient trop grand, la moyenne de la distance n'est plus faite que sur quelques points, ce qui cause de grandes fluctuations.

```

%% ANALYSE DU MOUVEMENT BROWNIEN
temps=0;
temps_max=size(center,1);
distance=[];
L=[];
pas=1;
for pas=1:1:16000;
    distance=[];
    for temps=(1+pas):pas:temps_max-pas
        if center(temps,1)==0 && center(temps,2)==0;

dist=((center(temps+pas,1)-center(temps-pas,1))^2+(center(temps+pas,2)-
center(temps-pas,2))^2)/2;
            distance=[distance dist];
        else

dist=((center(temps,1)-center(temps-pas,1))^2+(center(temps,2)-center(t
emps-pas,2))^2);
            distance=[distance dist];
            temps = 1+pas:pas:temps_max;
        end
    end
    L=[L mean(distance)];
end

```

```

figure, plot(log(1:1:16000),log(L(1:16000)))

```

Distance moyenne de diffusion dans une zone délimitée (crop):

On détermine la zone rectangulaire que l'on veut étudier en cliquant le point en haut à gauche et le point en bas à droite grâce à `getpts`, sur la première image de la série. Pour chaque position de la bille suivante, on regarde si elle se trouve dans la zone ou non et on augmente la distance de diffusion qui est en cours de calcul en conséquence. On stocke les distances parcourues à la suite dans la zone dans la variable `Lddif`, dont la moyenne donne la longueur moyenne de diffusion. Le paramètre `Normalisation`, égal à l'aire divisée par le périmètre de la zone d'intérêt, permet d'obtenir une longueur de diffusion normalisée que l'on pourra ensuite comparer avec les résultats sur des zones de diffusion de tailles différentes ou sur d'autres séries de données.

```
%% CROP, ANALYSE DES DONNEES
N1=size(center);
N=N1(1);
rep = 'E:\600b_15032019'; %Dossier dans lequel on trouve les images
ext = '*.tiff';
chemin = fullfile(rep,ext);
list = dir(chemin);
imgIni=imread(fullfile(rep, list(1).name), ext(3:end));
figure, imshow(imgIni)
[x,y] = getpts; %Prendre le point en haut à gauche puis celui en bas à
droite pour le crop
xmin=floor(x(1)); %ValeurPourLeCrop
ymin=floor(y(1));
xmax=floor(x(2));
ymax=floor(y(2));
l=0;
i=1;
for n=2:N-1;
    if
center(n,1)<xmax&(center(n,1)>xmin)&(center(n,2)<ymax)&(center(n,2)>ymin);
l=l+sqrt((center(n,1)-center(n-1,1))^2+(center(n,2)-center(n-1,2))^2);
    else
        L(i)=l;
        i=i+1;
        l=0;
    end
end
indice=find(L>0);
Lddif=L(indice);
%Lddif=sort(Lddif);
N2=size(Lddif);
N=N2(1);

% Affichage resultats
Moyenne=mean(Lddif) %longueur moyenne de diffusion
figure,hist(Lddif,80)
```

```
Normalisation=(ymax-ymin)*(xmax-xmin)/(2*(ymax-ymin+xmax-xmin)) %  
taille caracteristique du crop, aire/perimetre  
Lmoynorm=Moyenne/Normalisation %longueur moyenne de diffusion  
normalisee  
ecarttype=std(Lddif)/sqrt(N)  
  
title( 'Histogramme des longueurs de diffusion')  
xlabel('Longueur de diffusion en pixels') % x-axis label
```