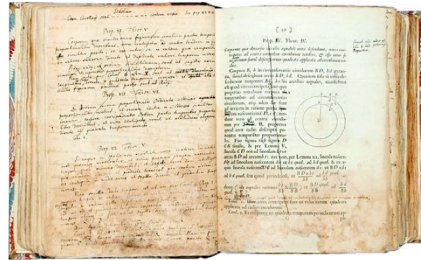
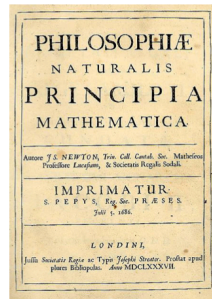


Matlab: applications en mécanique

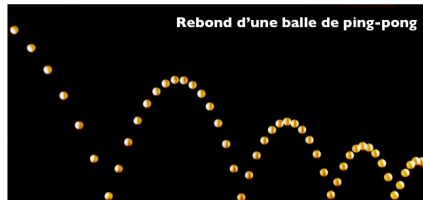
LA207, Université Pierre et Marie Curie.

2.3 TP3: Newton et la balle de ping-pong

Newton décrit dans les "principia", une des premières mathématisations d'une loi de la physique, le second principe de la dynamique, qui dit que un corps soumis à des forces est entraîné dans un mouvement: la somme des forces appliquées est égale à la masse fois l'accélération. Avec cette loi, une fois admis que les corps célestes exercent les uns sur les autres des forces d'attraction, on peut comprendre et prédire le mouvement des étoiles, des planètes et de leurs satellites. La balle de ping-pong elle aussi est soumise à cette loi. Entre les rebonds, la seule force appliquée est son poids, et pendant le rebond, la réaction du support joue. Il y a aussi d'autres forces, dont l'action est moins prépondérante: les efforts dus au déplacement de l'air: efforts aérodynamiques, et efforts de déformation de la balle pendant le rebond.



Isaac Newton:
les principes mathématiques de la philosophie naturelle
(la physique)




Dans ce TP, nous allons utiliser la photo du rebond de la balle de ping-pong comme une expérimentation physique pour vérifier la loi de Newton, et estimer les pertes énergétiques au cours du mouvement.

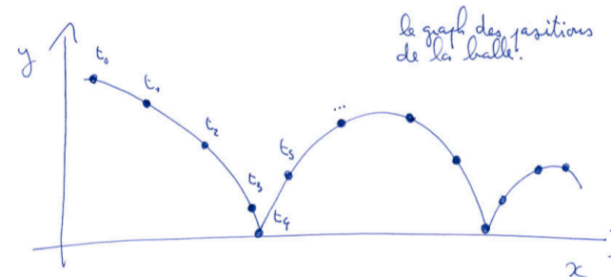
Compétences techniques:

- Mesurer la position de points sur une image.
- Calcul de la vitesse et de l'accélération à partir des positions successives.
- Tracer une trajectoire théorique, en prenant en compte les rebonds.

Données: Masse de la balle: $m=5$ grammes; largeur et hauteur d'un pixel de l'image: $dx=1.7$ millimètre; Intervalle entre les prises de vue: $dt=0.04$ secondes; Accélération de la gravité: $g=9.8$ mètres par seconde au carré.

2.3.1 Manipulations

1. L'image de la balle de ping-pong est stockée sur le disque avec pour nom de fichier: `pingpong.jpg`. Chargez la dans matlab avec la commande `imread`, puis affichez là, avec la commande `image`.
2. A l'aide de l'outil graphique d'étiquetage , mesurez sur l'image la position du centre de la balle aux temps successifs que vous stockerez dans deux tableaux `x` et `y`. Tracez le graphique de la trajectoire avec la fonction `plot`.



3. Calculez le vecteur vitesse: la vitesse selon x pour un temps donné, peut être calculée comme la position selon x au temps suivant moins la position selon x au temps précédent, divisé par l'intervalle de temps entre ces deux instants:

$$v_x(t) \approx \frac{x(t+dt) - x(t-dt)}{2dt}$$

Tracez le graph de la vitesse v_x selon x et v_y selon y en fonction du temps.

4. On calcule l'accélération. Pour la calculer, on se souviendra que l'accélération, c'est la "vitesse de la vitesse". Tracez le graph de l'accélération selon x et selon y en fonction du temps. A quoi est égale l'accélération verticale? A quoi est égale l'accélération horizontale? Que se passe-t'il lors des rebonds?

2.3.2 Etude

L'énergie mécanique de notre balle, c'est la somme de l'énergie cinétique $E_c = m\|v\|^2/2$ et de l'énergie potentielle $E_p = mgy$. Au cours du mouvement, l'énergie peut être transférée de l'énergie cinétique vers l'énergie potentielle et réciproquement, par contre, s'il n'y a pas de frottement, l'énergie totale reste constante.

Tracez un graph qui montre comment les énergies varient dans le temps. Grâce à ce graph, analysez les pertes d'énergie de notre balle. Pour cela, posez-vous les questions: l'énergie est-elle constante ou pas? Quels sont les effets physiques qui peuvent faire varier l'énergie? Que se passe-t'il lors du rebond?

Vérifiez que le coefficient α de restitution d'énergie lors du rebond est le même pour tous les rebonds: énergie après le rebond= α fois l'énergie avant le rebond.

2.3.3 Pour aller plus loin

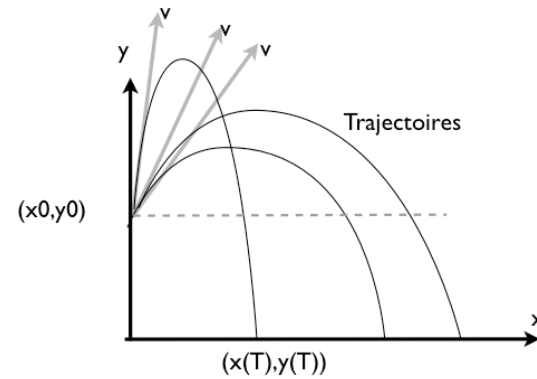
Nous voulons maintenant comparer la trajectoire mesurée avec une trajectoire théorique qui obéit à la loi de Newton. On va prendre en compte la perte d'énergie pendant le rebond, mais pas la résistance aérodynamique.

Selon la loi de Newton, la trajectoire est

$$x(t) = x_0 + v_x t, \quad y(t) = y_0 + v_y t - gt^2/2$$

avec (x_0, y_0) et (v_x, v_y) les position et vitesse initiales de la balle. Ecrivez une fonction `rebondfnc` qui trace la trajectoire de la balle pour (x_0, y_0) et (v_x, v_y) donnés. Cette fonction donnera en arguments de sortie le temps T auquel la balle touche le sol ($y(T) = 0$) et la vitesse $(v_x(T), v_y(T))$ et la position de la balle $(x(T), y(T))$ à cet instant T .

Tracez sur un graphique la superposition des trajectoires pour $x_0 = 0, y_0 = 1$, et $\|v\| = 10$, ou la vitesse initiale fait un angle θ avec l'horizontale. Vous prendrez plusieurs valeurs pour θ entre $\pi/2$ et $-\pi/2$.

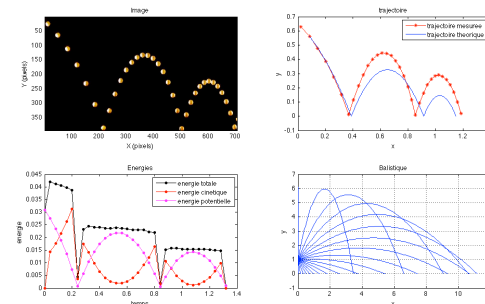


Comparaison mesures/théorie: lors d'un rebond, la vitesse change comme ceci:

$$v_x \rightarrow \alpha \times v_x, \quad v_y \rightarrow -\alpha \times v_y,$$

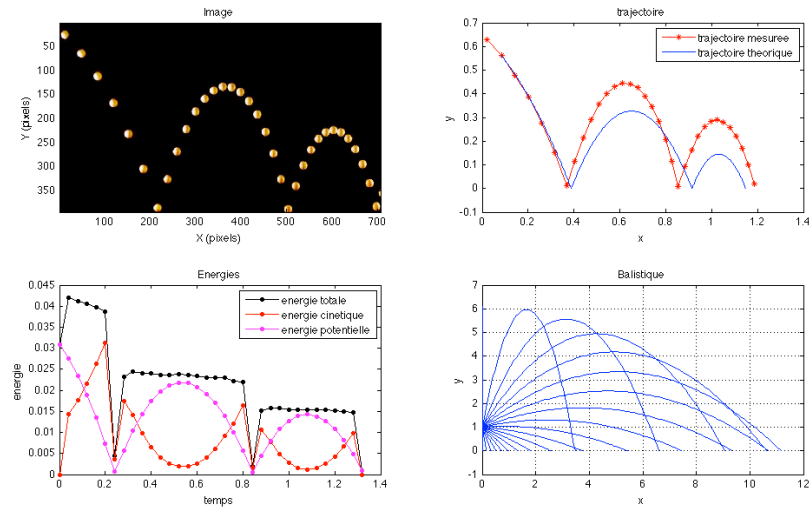
c'est à dire, les deux composantes de la vitesse sont multipliées par le facteur de restitution α , et la vitesse selon y change de signe. Utiliser votre fonction `rebondfnc` pour comparer la trajectoire mesurée avec la trajectoire théorique en prenant un facteur de restitution $\alpha = 2/3$.

Pour finir, créez une figure sur laquelle on voit tous les résultats à la fois avec des `subplot`: l'image expérimentale, la trajectoire mesurée comparée avec la trajectoire théorique, l'évolution des énergies et les trajectoires balistiques avec la fonction `rebondfnc`.



TP3: correction

Sur la figure ci-dessous, il y a tous les résultats attendus. Le graph en bas à gauche est celui qu'on obtient à la fin de la partie "Etude", lors de la partie "Pour aller plus loin", on obtient le graph en bas à droite des trajectoires balistiques, et la courbe bleue sur le graph en haut à droite.



Cette figure est générée par le script suivant:

```
% pour le TP du rebond de la balle
clear
clf

% paramètres
dx=0.0017;
dt=0.04;
ymax=394;
m=0.005;
g=9.81;

% affichage de l'image
subplot(2,2,1);
a=imread('pingpong.jpg');
image(a)
xlabel('X (pixels)'); ylabel('Y (pixels)'); title('Image')
```

```
% les points de mesure
x=[12, 49, 84 120 153 185 216 238 259 279 299 ...
  320 339 360 380 399 418 436 454 471 487 ...
  503 520 537 553 570 586 602 618 635 651 667 682 697];
y=[25 64 112 167 232 305 386 326 269 222 ...
  185 158 141 132 134 144 164 190 228 273 326 ...
  389 340 298 264 241 228 223 229 242 264 295 335 383];

% mise à l'échelle
x=dx*x;
y=dx*(ymax-y);

% on trace la trajectoire
subplot(2,2,2);
plot(x,y,'r*');
xlabel('x'); ylabel('y'), title('trajectoire')

n=length(x);
tvec=0:dt:(n-1)*dt;

% calcul de la vitesse
vx=zeros(1,n);
vy=zeros(1,n);
for ind=2:n-1
    vx(ind)=(x(ind+1)-x(ind-1))/(2*dt);
    vy(ind)=(y(ind+1)-y(ind-1))/(2*dt);
end

% calcul de l'accélération
ax=zeros(n,1);
ay=zeros(n,1);
for ind=3:n-2
    ax(ind)=(vx(ind+1)-vx(ind-1))/(2*dt);
    ay(ind)=(vy(ind+1)-vy(ind-1))/(2*dt);
end

% energie cinétique et potentielle
ec=m*(vx.^2+vy.^2)/2;
ep=m*y*g;

% on trace les energies
subplot(2,2,3)
plot(tvec,ec+ep,'k.-',tvec,ec,'-r',tvec,ep,'m.-')
legend('energie totale','energie cinétique','energie potentielle');
```

```

xlabel('temps'); ylabel('energie'); title('Energies')

%%%%%%%%%% trajectoires balistique:
% avec la fonction rebondfonc.m
subplot(2,2,4);
thetavec=linspace(-pi/2,pi/2,20);
v=10;
for ind=1:length(thetavec);
    theta=thetavec(ind);
    vvx=v*cos(theta);
    vvy=v*sin(theta);

    x0=0;
    y0=1;
    [x0,vvx,vvy,tmax]=rebondfonc(x0,y0,vvx,vvy);

    hold on
end
grid on
xlabel('x'); ylabel('y'); title('Balistique')

%%%%%%%%%% pour comparer la courbe théorique
% avec les mesures
subplot(2,2,2);
hold on

% position et vitesse initiale de notre balle
% prises à la position loc
loc=2;
y0=y(loc);
x0=x(loc);
vvx=vx(loc);
vvy=vy(loc);

% coef de restitution
resti=2/3;

% première parabole
[x0,vvx,vvy,tmax]=rebondfonc(x0,y0,vvx,vvy);
hold on

% rebond et seconde parabole

```

```

vvx=resti*vvx;
vvy=-resti*vvy;
[x0,vvx,vvy,tmax]=rebondfonc(x0,0,vvx,vvy);

% rebond et troisième parabole
vvx=resti*vvx;
vvy=-resti*vvy;
[x0,vvx,vvy,tmax]=rebondfonc(x0,0,vvx,vvy);

legend('trajectoire mesurée','trajectoire théorique')

% couleur de la figure en blanc plutôt que gris
set(gcf,'color','w')

    Et on utilise aussi la fonction rebondfonc.m:

function [x0,vx,vy,tmax]=f(x0,y0,vx,vy)
% pour le TP du rebond de la balle
g=9.8;

% le temps final du rebond
tmax=(vy+sqrt(vy^2+4*y0*g/2))/g;
t=linspace(0,tmax,100);

% la trajectoire
x=vx*t+x0;
y=-g*t.^2/2+t*vy+y0;

% vitesse et position finales
vy=-g*tmax+vy;
x0=x0+vz*tmax;

% on trace
plot(x,y);

```