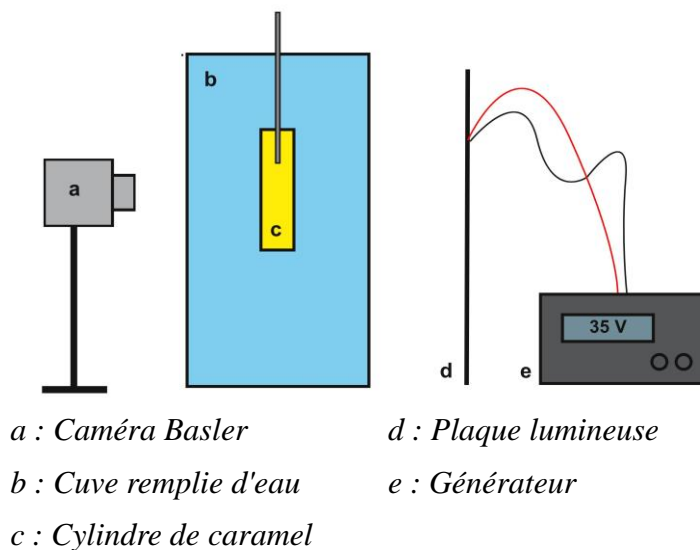


Matériel et Méthodes

I. Matériel

- *Préparation du caramel :*
Casserole
Balance
Plaque Chauffante
Spatule en bois
Sucre et Sirop de glucose
Moule en silicone (cf onglet *Moule*)
Tiges en acier (diamètre 4,03 cm)
- *Dissolution du caramel :*
Cuve en plexiglas (dimension 20,8X20,8X40,5 cm³)
Particules de PIV PM 50-100 µm
- *Acquisition d'images :*
Caméra : Basler, aCA 1300-200µc, ID : 106754-14 ; S/N : 22502719
Alimentation : Elektro-Automatik EA-PS 2042-06B
Plaque lumineuse (à environ 35 V)
Nappe laser : Laser Module TZ 12X50-520L, Class IIIa, 520 nm (alimenté à 3V)
- *Montage expérimental :*



II. Protocoles

1) Recette du caramel

Ingrédients :

- 100g de sucre (Saint Louis)
- 50g de sirop de glucose (Top Cake)

On mélange le sucre et le sirop de glucose dans une casserole et on met à chauffer.

Chauffage :

-100°C pendant 10 min ;

-120°C pendant 5 min ;

-140°C pendant 10 min ;

-160°C pendant 18 min (ce temps peut être modifié en fonction de la couleur de caramel souhaitée).

Ne pas mélanger pendant l'étape de chauffage sauf s'il reste du sucre sur les bords.

Une fois les étapes de chauffage terminées, retirer le caramel du feu et mélanger doucement.

Couler le caramel dans un moule, insérer une tige si nécessaire et attendre que le caramel refroidisse.

2) Moule

Mélange 50/50 de :

- Elite Double 32 (vert clair) vinylpolysiloxane (A-silicone) Zhermack
- Elite Double 22 (blanc) vinylpolysiloxane (A-silicone) catalyst Zhermack

Mise en forme jusqu'à formation d'un moule stable

Dimensions : cylindre de hauteur 5,25 cm et de diamètre 2,2 cm

3) Dissolution

Remplir la cuve d'eau à température ambiante. Deux protocoles de dissolution sont alors élaborés :

- *Dissolution avec observation de l'évolution de l'aspect du caramel*
Allumer la plaque lumineuse et régler l'acquisition d'images de la caméra à 0,1 Hz avec le logiciel *PylonViwer* et régler la netteté et la luminosité de l'image. Plonger le caramel dans la cuve et attendre la dissolution complète (durée d'environ 1h30).
- *Dissolution avec observation des flux (PIV)*
Se placer dans le noir et éclairer la cuve perpendiculairement à la caméra avec la nappe laser. Ensemencer l'eau de la cuve avec des microparticules de PIV (environ 2 spatules). Régler la fréquence d'acquisition de la caméra à 10 Hz. Placer le caramel dans la cuve afin qu'il coupe le plan du laser. Régler la netteté de la caméra. Attendre que l'équilibre s'installe dans la cuve puis lancer l'acquisition jusqu'à obtenir environ 3000 images.

4) Traitement d'images avec Image J

- Ouverture d'une séquence d'images sur ImageJ
- Passage en 8 bits

- Image > Adjust > Threshold : ajustement du seuil
- Process > Binary > Fill Holes puis Dilate puis Erode : correction des potentiels trous
- Analyse Set Measurements : cocher l'aire pour que le logiciel la calcule
- Analyse Particles : obtention de l'aire du caramel sur chaque image de la séquence, mettre un seuil (par exemple 5000) pour enlever les petites particules parasites

III. Programmes informatiques

1) Programmes Matlab

Programme de calcul des distances entre les bords inférieur et supérieur et la tige.

Entrée du programme : série d'images d'une dissolution préalablement traitée avec le protocole ImageJ. Les images ont été enregistrées à une fréquence de 0,1 Hz.

Sortie du programme : Courbes de l'évolution des distances moyennes entre les bords supérieur et inférieur et la tige au cours du temps.

```

N=300;
f=rdir('NB\*.tif');
A=imread(f{50});
B=edge(A);
imagesc(1-B);
% Tracé de la ligne milieu et distance du contour à cette droite
[X,Y]=getline(gcf);
%a=coef directeur b=ordonnée à l'origine
a=(Y(2)-Y(1))/(X(2)-X(1));
b=Y(1)-a*X(1);
%[W,Z]=getline(gcf);

%ligne limite
[W,Z]=getline(gcf);
a2=(Z(2)-Z(1))/(W(2)-W(1));
b2=Z(1)-a2*W(1);

%deuxième ligne limite
[U,V]=getline(gcf);
a3=(V(2)-V(1))/(U(2)-U(1));
b3=V(1)-a3*U(1);

Minf60=[];
Msup60=[];

for (i=1:N);
    A=imread(f{i});
    B=edge(A);
    imagesc(1-B);
    Y1=[1:size(B,1)];
    X1=[1:size(B,2)];

```

```

[X2,Y2]=meshgrid(X1,Y1);
D=abs(Y2-a.*X2-b)./sqrt(1+a*a).*B;
%on coupe B
B1 = (a2*X2 - Y2 + b2). *B;
B1 = (-((sign(B1)-1))/2).*B;

B2 = (a3*X2 - Y2 + b3). *B1;
B2 =((sign(B2)+1)/2).*B1;

%matrice des distances de la ligne de dessous par rapport à la ligne milieu
Binf= (a*X2 - Y2 + b). *B2;
Binf=(-((sign(Binf)-1))/2).*B2;

Dinf=D.*Binf;
D1=Dinf;
p=find(Dinf==0);
D1(p)=[];
moy1=mean(D1);
Minf60(i)=moy1;

%matrice des distances de la ligne du dessous par rapport à la ligne milieu
Bsup= (a*X2 - Y2 + b). *B2;
Bsup=((sign(Bsup)+1)/2).*B2;

Dsup=D.*Bsup;
D2=Dsup;
p2=find(Dsup==0);
D2(p2)=[];
Msup60(161)=Msup60(161);
for (i=162:N);
    Minf60(i)=Minf60(i);
    Msup60(i)=Msup60(i);
    moy2=mean(D2);
    Msup60(i)= moy2;
end
Minf60(161)=Minf60(161);
end
Minf60 = Minf60 * 1.3735e-1;
Msup60 = Msup60 * 1.3735e-1;

Temps=linspace(0,N*10,N);
plot(Temps,Minf60,'b')
title('Evolution des distances bord inférieur/bord supérieur fonction du temps')
xlabel('Temps (s)')
ylabel('Distance (mm)')
hold on
plot(Temps,Msup60,'r')
legend('Bord inférieur','Bord supérieur')

```

Programme réalisant la PIV : module OpenPIV de Matlab.

Entrée du programme : Série d'images montrant le déplacement de particules en suspension autour du caramel, éclairées par une nappe laser. Les images sont enregistrées à une fréquence de 10 Hz. Le programme fonctionne par corrélation d'images. Il faut donc choisir la taille d'un élément de base à corrélérer (choix dans le projet : 34 * 34 pixels).

Sortie du programme : Fichier de données contenant la position (x et y) et la vitesse (dans les deux directions) de chaque élément au cours du temps.

Programme traçant le champ de vitesse à partir des images préalablement traitées par le module openPIV.

Entrée du programme : Fichier de données obtenu par le module OpenPIV.

Sortie du programme : Image représentant le champ des vitesses des particules autour du caramel.

```
scale=1;
xG=0;
xD=600;
yB=0;
yU=1024;
delimiterIn = ' ';
headerlinesIn = 1;
nblm=55; %length(dir('.')/4-1;
X=[];
Y=[];
UX=[];
UY=[];
for i=0:54 %nblm-1
filename=['gouter 18' num2str(i,'%4d')];
A = importdata([filename '_flt.txt'],delimiterIn,headerlinesIn);
data=A.data;
figure(1)
X = [X ; data(:,1)'];
Y = [Y ; data(:,2)'];
UX = [UX ; data(:,3)'];
UY = [UY ; data(:,4)'];
im=imread([filename '.png']);
imagesc(im);
hold on
quiver(data(:,1),data(:,2),data(:,3),data(:,4),scale,'w');
axis([xG xD yB yU])
axis image
title([filename '_flt.txt']);
hold off
```

2) Programmes Python

Programme donnant l'évolution de l'aire du cylindre en fonction du temps.

Entrées du programme : Fichiers texte contenant les aires des cylindres au cours de la dissolution. Ces fichiers sont créés par le protocole ImageJ.

Sortie du programme : Droite représentant l'évolution de l'aire du cylindre en fonction du temps pour les différents angles.

```
f2=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 2/aire
2.txt","r")
f5=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 5/aire
5.txt","r")
f6=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 6/aire
6.txt","r")
f7=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 22/aire
22.txt","r")
f11=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 11/aire
11.txt","r")
f25=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 25/aire
25.txt","r")
f29=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 29/aire
29.txt","r")
```

```
c=0
l2=[]
for l in f2 :
    c= l.split('\n')
    c=float(c[0])
    l2.append(c)
m2=max(l2)
l2=[i/m2 for i in l2]
n2=len(l2)
X2=[]
for i in range(n2):
    X2.append(i*10)
```

```
c=0
l5=[]
for l in f5 :
    c= l.split('\n')
    c=float(c[0])
    l5.append(c)
n5=len(l5)
m5=max(l5)
l5=[i/m5 for i in l5]
X5=[]
for i in range(n5):
    X5.append(i*10)
```

```

c=0
l6=[]
for l in f6 :
    c= l.split('\n')
    c=float(c[0])
    l6.append(c)
n6=len(l6)
m6=max(l6)
l6=[i/m6 for i in l6]
X6=[]
for i in range(n6):
    X6.append(i*10)

```

```

c=0
l11=[]
for l in f11 :
    c= l.split('\n')
    c=float(c[0])
    l11.append(c)
n11=len(l11)
m11=max(l11)
l11=[i/m11 for i in l11]
X11=[]
for i in range(n11):
    X11.append(i*10)

```

```

c=0
l29=[]
for l in f29 :
    c= l.split('\n')
    c=float(c[0])
    l29.append(c)
n29=len(l29)
m29=max(l29)
l29=[i/m29 for i in l29]
X29=[]
for i in range(n29):
    X29.append(i*10)

```

```

c=0
l25=[]
for l in f25 :
    c= l.split('\n')
    c=float(c[0])
    l25.append(c)
n25=len(l25)
m25=max(l25)
l25=[i/m25 for i in l25]
X25=[]
for i in range(n25):

```

```

X25.append(i*10)

c=0
l7=[]
for l in f7 :
    c= l.split('\n')
    c=float(c[0])
    l7.append(c)
n7=len(l7)
m7=max(l7)
l7=[i/m7 for i in l7]
X7=[]
for i in range(n7):
    X7.append(i*10)

plt.plot(X2,l2,'b-',label ="0°")
plt.plot(X6,l6,'k-',label="13°")
plt.plot(X5,l5,'r-',label="40°")
plt.plot(X11,l11,'g-',label="62°")
plt.plot(X29,l29,'y-',label="70°")
plt.plot(X25,l25,'c-',label="80°")
plt.plot(X7,l7,'m-',label="90°")
plt.xlim((0, 4500))
plt.ylim((0.1 , 1))
plt.legend()
plt.ylabel('Aire normalisée')
plt.xlabel('T (s)')
plt.title("Evolution de l'aire en fonction du temps")
plt.show()

```

Programme représentant la valeur de la pente de l'évolution de l'aire en fonction de l'angle du cylindre.

Entrées du programme : Fichiers texte contenant les aires des cylindres au cours de la dissolution. Ces fichiers sont obtenus par le protocole ImageJ.

Sortie du programme : Graphique représentant la valeur absolue de la pente des droites de l'évolution de l'aire en fonction du temps, en fonction de l'angle.

```

from matplotlib import pyplot as plt
import scipy as sc
from scipy import stats
import numpy as np
f2=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 2/aire
2.txt","r")
f5=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 5/aire
5.txt","r")
f6=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 6/aire
6.txt","r")
f7=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 22/aire
22.txt","r")

```



```
f11=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 11/aire
11.txt","r")
f25=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 25/aire
25.txt","r")
f29=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 29/aire
29.txt","r")
```

```
c=0
l2=[]
for l in f2 :
    c= l.split('\n')
    c=float(c[0])
    l2.append(c)
l2=l2[:200]
```

```
c=0
l5=[]
for l in f5 :
    c= l.split('\n')
    c=float(c[0])
    l5.append(c)
l5=l5[:200]
```

```
c=0
l6=[]
for l in f6 :
    c= l.split('\n')
    c=float(c[0])
    l6.append(c)
l6=l6[:200]
```

```
c=0
l7=[]
for l in f7 :
    c= l.split('\n')
    c=float(c[0])
    l7.append(c)
l7=l7[:200]
```

```
c=0
l11=[]
for l in f11 :
    c= l.split('\n')
    c=float(c[0])
    l11.append(c)
l11=l11[:200]
```

```

c=0
l25=[]
for l in f25 :
    c= l.split('\n')
    c=float(c[0])
    l25.append(c)
l25=l25[:200]

X=[]
for i in range(200):
    X.append(i*10)

```

```

c=0
l29=[]
for l in f29 :
    c= l.split('\n')
    c=float(c[0])
    l29.append(c)
l29=l29[:200]

```

```

X=[]
for i in range(200):
    X.append(i*10)

```

```

pente=[]
lr2=sc.stats.linregress(X,l2)
pente.append(lr2[0])
lr6=sc.stats.linregress(X,l6)
pente.append(lr6[0])
lr5=sc.stats.linregress(X,l5)
pente.append(lr5[0])
lr11=sc.stats.linregress(X,l11)
pente.append(lr11[0])
lr29=sc.stats.linregress(X,l29)
pente.append(lr29[0])
lr25=sc.stats.linregress(X,l25)
pente.append(lr25[0])
lr7=sc.stats.linregress(X,l7)
pente.append(lr7[0])

```

```
angle=[0,13,40,62,70,80,90]
```

```

pente=[-i for i in pente]
plt.plot(angle,pente,'Dm')
plt.ylabel('| pente |')
plt.xlabel('angle ( ° )')
plt.title("|Pente| en fonction de l'angle")
plt.xlim((-2, 92))

```

```
plt.show()
```

Programme représentant le contour du cylindre à différents temps.

Entrée du programme : Fichier contenant la position des contours du cylindre de caramel pour chaque image prise pendant la dissolution. Ce fichier est obtenu par le protocole ImageJ.

Sortie du programme : Graphique représentant le contour du cylindre à différents instants de la dissolution.

```
from matplotlib import pyplot as plt
f=open("/Users/laetitiafurno/Documents/Travail/ESPCI 1A/PSE/Gouter 22/contour
22.txt","r")
```

```
c1=0
c2=0
c3=0
c4=0
```

```
l1=[]
l2=[]
l3=[]
```

```
for l in f :
    c1,c2,c3,c4 = l.split('\t')
    c1=int(c1)
    c2=int(c2)
    c3=int(c3)
    l1.append(c1)
    l2.append(c2)
    l3.append(c3)
```

```
#Création des coordonnées pour une image n (n partant de 0)#
```

```
def coord(l1,l2,l3,n):
    X=[]
    Y=[]
    i=0
    while(l3[i]!=n):
        i=i+1
    while(l3[i] == n):
        X.append(l1[i])
        Y.append(l2[i])
        i=i+1
    T=[X,Y]
    return(T)
```

```
plt.plot(coord(l1,l2,l3,0)[0],coord(l1,l2,l3,0)[1],'b.',label="0 s")
plt.plot(coord(l1,l2,l3,100)[0],coord(l1,l2,l3,100)[1],'m.',label="1000 s")
plt.plot(coord(l1,l2,l3,200)[0],coord(l1,l2,l3,200)[1],'c.',label="2000 s")
plt.plot(coord(l1,l2,l3,350)[0],coord(l1,l2,l3,350)[1],'k.',label="3500 s")
plt.legend()
```

```
plt.ylabel('Y')  
plt.xlabel('X')  
plt.xlim(0,500)  
plt.title('Contours 62 °')  
plt.show()
```

Laetitia Furno, Rémi Gaillard et Agathe Legendre