

MATÉRIEL ET MÉTHODES
Information par train de gouttes

Laure Le Blanc, Aliénor Lahlou, Victoria Vilchez

13/05/2017

Introduction

Le problème traité ici s'inscrit dans le cadre scolaire d'un projet scientifique en équipe (PSE) réalisé par trois élèves de l'ESPCI Paris (Laure Le Blanc, Aliénor Lahlou et Victoria Vilchez). L'objectif est de réaliser un système microfluidique capable de crypter un message codé sur des gouttelettes, puis de tenter de le décrypter.

1 Réalisation de canaux en microfluidique

1.1 Réalisation du template

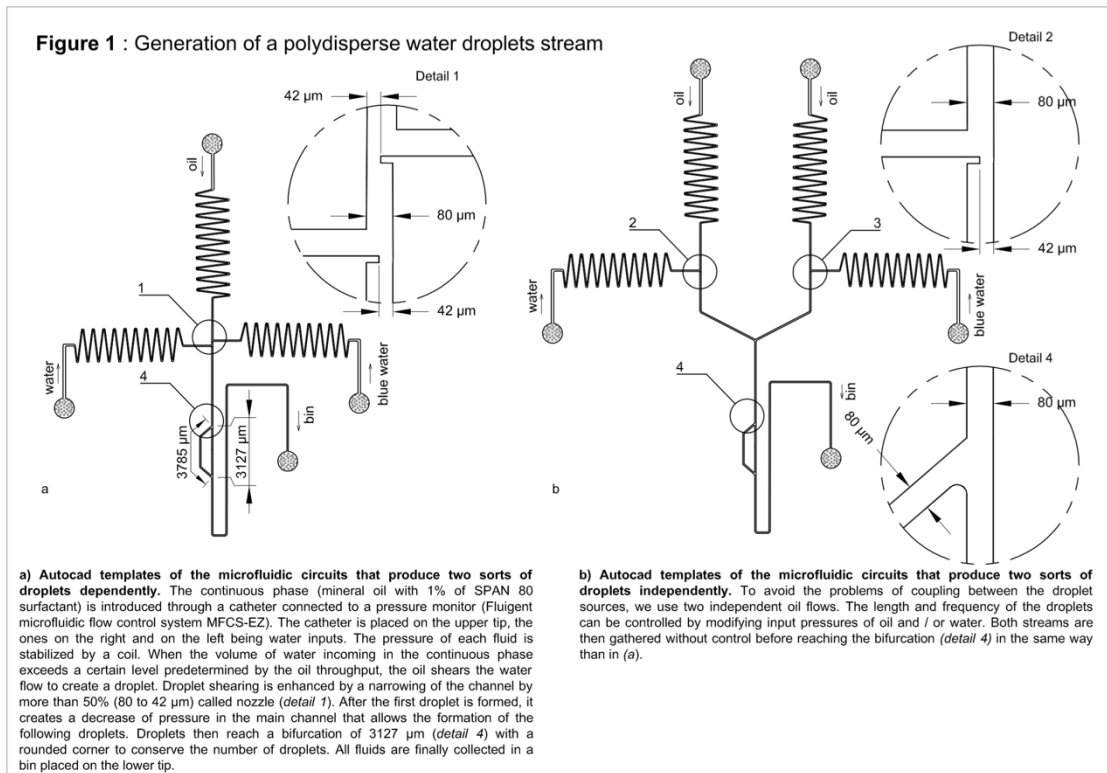
On trace sous Autocad 2017, logiciel de dessin assisté par ordinateur (DAO), le template ou "design" du canal souhaité. Notre objectif est de former des séquences de deux types de gouttes (colorée et incolore) dans l'huile.

Au cours de ce projet, on décide de tester deux configurations de microsystèmes. Dans un premier temps, la formation des deux types de gouttes est couplée. Tandis que dans un second temps, nous découplons la formation des deux types de gouttes en introduisant un flux d'huile supplémentaire.

On modélise plusieurs entrées de fluides (huile+SPAN80 à 1%, eau incolore, eau colorée). A chaque entrée, on trace un serpentin afin de stabiliser la pression qui sera appliquée. Les canaux sont tracés de sorte à avoir une largeur de $80\ \mu\text{m}$. On rétrécit de moitié le canal de sortie (largeur $40\ \mu\text{m}$) en dessinant un nozzle à chaque embranchement des voies d'entrées afin d'optimiser la formation des gouttes.

Dans la partie du canal où les gouttes sont formées, on introduit un mélange. Pour forcer passivement le mélange, on induit une différence de résistance hydraulique entre les deux branches en leur donnant des longueurs différentes. Le mélange est ensuite conduit dans une poubelle.

Une fois l'intégralité du canal dessiné, on vérifie que l'ensemble des traits forme une figure fermée afin de confiner les fluides. Si ce n'est pas le cas, on joint bien un à un chacun des traits à l'endroit qui fait défaut.



1.2 Réalisation du masque

L'ensemble des canaux tracés est délimité par une fenêtre type, dont les côtes sont déterminées par le Laboratoire de Biochimie de l'ESPCI Paris. Celle-ci permet à l'imprimante de reproduire le template sur un masque que l'on récupère.

1.3 Réalisation du wafer

La réalisation du wafer s'effectue par photolithographie en salle blanche à l'Institut Pierre Gilles de Gennes, avec blouse, charlotte et gants.

Une plaque de silicium 4 pouces est maintenue par du vide sur le plateau tournant à haute vitesse constante d'un spin coater, afin d'étaler la résine de façon uniforme par force centrifuge. On dépose une noisette de résine photosensible SU-8-2050 sur la plaque de silicium déposée sur le plateau à l'arrêt, de sorte à ce qu'il y en ait suffisamment pour recouvrir intégralement la plaque. On fait tourner le plateau à 3000 tour/min pour obtenir une épaisseur de 60 à 80 μm , pendant 1 min. On fait préchauffer l'ensemble plaque de silicium + résine, prélevé à l'aide d'une pince, à 65°C pendant 3 min, puis chauffer à 95°C pendant 9 min afin d'évaporer progressivement l'excès de solvant présent dans la résine tout en évitant les craquements et les inhomogénéités. Dans une machine UV-Cube, on introduit dans l'ordre : la plaque de silicium, le masque dont le côté délébile est adjacent à la plaque en silicium, puis une plaque en quartz afin de gagner en résolution. On expose l'ensemble pendant 9 sec à puissance maximale, soit 23,4 mW.cm⁻¹. Seule la résine exposée réticule. Post-exposition, on fait chauffer à nouveau l'ensemble à 65°C pendant 2 min, et à 95°C pendant 7 min. Puis on rince l'excès de résine non réticulée soluble avec un liquide développeur.

1.4 Coulage du polydiméthylsiloxane (PDMS)

En salle grise, on ajoute 5mL de réticulant dans 45g de sylgard. On mélange jusqu'à obtenir dans tout le volume des bulles homogènes. On introduit le mélange dans un tube Ependorff, que l'on passe pendant 5 min à la centrifugeuse pour que le mélange devienne uniforme, sans bulles. On dépose le wafer sur un moule en papier aluminium placé sur une boîte de Pétri. On coule délicatement le PDMS sur le wafer en évitant la reformation de bulles, puis on place la boîte sous une cloche à vide afin que les dernières bulles remontent à la surface et disparaissent. Une fois que toutes les bulles ont disparu, on place la boîte de Pétri fermée et scotchée à l'étuve à 70°C environ, pendant 1h.

1.5 Mise en forme du PDMS

On détache minutieusement le PDMS polymérisé du wafer. En salle grise, on découpe le PDMS de sorte à pouvoir placer les microsystèmes sur une lame de microscope. On veille à ne pas salir la partie ouverte des canaux. On perce les endroits du PDMS où les fluides seront injectés afin d'accéder au microsystème. On scotche ensuite ces trous pour ne pas que les poussières de l'environnement extérieur pénètrent dans les canaux. On rince le PDMS (après avoir ôté le scotch) et les lames à utiliser à l'isopropanol. On laisse sécher à l'étuve à 70°C environ pendant 1h.

1.6 Collage du PDMS sur une lame en verre

On utilise un traitement plasma de dioxygène pour modifier les propriétés de surface du PDMS. Ce traitement permettra de coller le microsystème sur une lame de verre. Les deux surfaces sont placées dans le plasma pendant 30 sec. L'oxydation des surfaces permet d'activer les deux surfaces. Les groupements $-\text{O}-\text{Si}-(\text{CH}_3)_2$ présents à la surface du PDMS génèrent des groupements silanols $-\text{O}-\text{Si}-\text{OH}$ qui se substituent aux groupements méthyles $-\text{CH}_3$. Cette réaction chimique permet de faire fortement adhérer les surfaces entre elles, par interaction électrostatique.

1.7 Silanisation

On rend les parois des canaux hydrophobes par injection d'une huile fluorée HFE et d'un silane afin de permettre la formation de gouttes d'eau dans l'huile. On sèche ensuite les canaux en injectant de l'azote.

2 Protocole expérimental

2.1 Mise en place du dispositif expérimental

L'injection de fluides dans le microsysteme peut être contrôlé en débit en utilisant des pousserieseringues, en pression par gravité, ou en utilisant un contrôleur de pression. On choisit d'utiliser un contrôleur de pression Fluigent MFCS-EZ afin d'accéder précisément et de manière stable à une plus large gamme de pression tout en ayant un temps de réponse à une variation de consigne en temps réel. Le contrôleur comporte quatre voies indépendantes, reliées au microsysteme par des tubes en polymère. On relie au contrôleur un échantillon d'eau incolore, un d'eau contenant quelques gouttes de bleu de méthylène, et deux échantillons contenant de l'huile minérale et du surfactant SPAN 80 à 1%. On connecte un fil à chaque entrée de fluide à l'aide d'un cathéter. On injecte dans un premier temps uniquement la phase continue (huile minérale à 1% en SPAN 80) dans le microsysteme. On visualise par la suite la formation de gouttes au microscope Leica DM4M. Des séquences vidéos sont capturées au niveau du mélangeur, puis analysées.

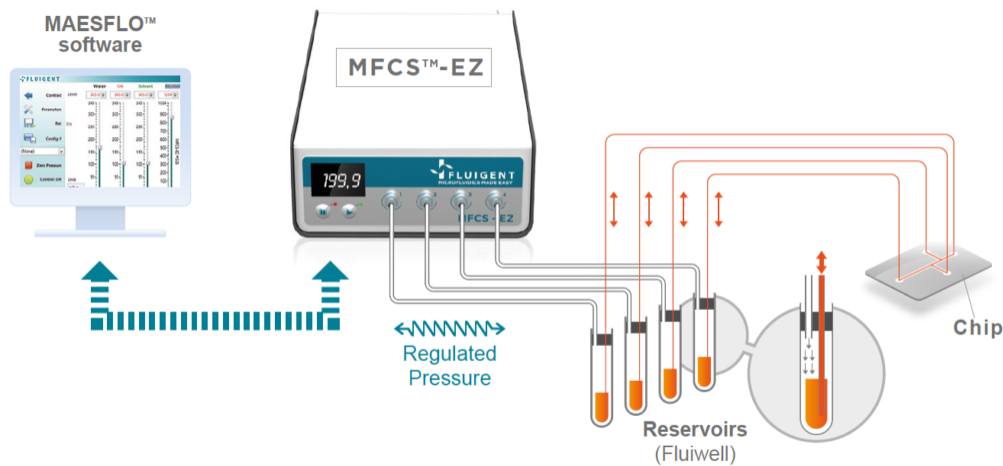


FIGURE 1 – Dispositif expérimental (source : fluigent.com)

Paramètres utilisés

	Formation de gouttes indépendantes	Formation de gouttes couplées
Pression d'eau incolore (mbar)	69,35	139,0
Pression d'eau colorée (mbar)	110,9	138,0
Pression huile+SPAN80 (mbar)	[136,8 ; 190,0]	149,2

2.2 Mélanger des gouttes

La séquence de goutte initiale est modifiée à l'embranchement de type jonction T avec le mélangeur. Le mélangeur contient un canal inférieur de $3127 \mu\text{m}$ et un canal supérieur de $3785 \mu\text{m}$ avec un angle de 49° . Lors de son arrivée à l'embranchement, une goutte entre dans le canal dont la résistance hydraulique est la plus faible. Cette résistance hydraulique est variable au cours de l'expérience puisque chaque goutte d'eau a une résistance propre, différente localement de celle du canal lorsqu'il contient de l'huile. Cette différence de résistance vient des différences physico-chimiques entre les liquides (viscosité dynamique notamment), et des frottements eau/huile aux interfaces. Chaque goutte entrant dans un canal accroît la résistance hydraulique de celui-ci, d'après l'équivalence entre la résistance d'une goutte et la résistance d'une portion de longueur L de canal. La longueur L dépend de la taille de la goutte.¹ Ainsi, à partir d'un certain nombre de gouttes dans le canal supérieur, la résistance dans cette branche dépasse celle du canal inférieur. Les gouttes arrivant à l'embranchement à cet instant choisiront alors le canal inférieur, et réciproquement. La séquence de gouttes se trouve donc modifiée.

1. Droplet Traffic at a Simple Junction at Low Capillary Numbers, W. Engl, M. Roche, A. Colin, and P. Panizza *Phys. Rev. Lett.* **95**, 208304 (2005)

3 Traitement des données

3.1 Récupération du train du gouttes

On traite les images obtenues avec un module de traitement d'image, SMIL (Simple Morphological Image Library) développé par le CMM aux Mines de Paris. On traite en Python les images par des opérations morphologiques mathématiques. On se sert aussi de Python pour traiter les données obtenues avec le traitement d'image.

Nous avons rédigé un programme pour extraire la forme et la position des gouttes des circuits. Il fonctionne à condition de prendre un « blanc » du circuit : une photo du circuit rempli d'huile et sans goutte. Ce programme, « extraction.py » permet d'obtenir des gouttes qui apparaissent en zones uniformes blanches facilement détectables et labelisables. A la fin, on superpose toutes les gouttes détectées sur une même image pour repérer le lieu de passage des gouttes et vérifier que le programme n'a pas fait d'erreurs (en confondant des poussières avec des gouttes par exemple).

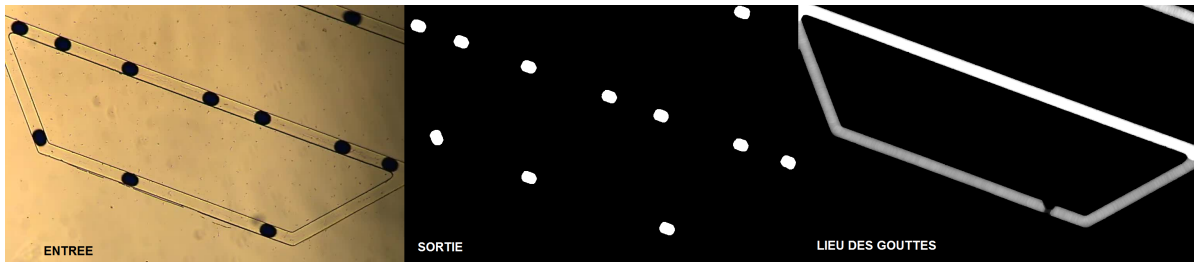


FIGURE 2 – A gauche : 1 image d'une vidéo en entrée du programme. Au centre : 1 image d'une vidéo en sortie. A droite : lieu de passage de l'ensemble des gouttes

3.2 Analyse des images traitées

Pour les propriétés que nous étudions, nous avons besoin de connaître les chemins choisis par les gouttes. Nous avons écrit un programme, « analyse.py », qui procède par un système de balises. On a 5 balises, et pour chaque image consécutive, on attribue la valeur 1 à chaque balise sur laquelle se trouve une goutte, et 0 sinon. Nous avons écrit le programme pour accepter n'importe quel type de balise, sous forme d'une image entière qu'on compare à l'image avec les gouttes. Cependant, pour ce dont nous avons besoin, les balises sont des images noires avec un pixel blanc qui représente la balise. Cela revient donc à regarder la valeur de 5 pixels sur chaque image traitée préalablement avec extraction.py, et voir s'ils sont blanc (goutte -> balise = 1) ou noir (pas de goutte -> balise = 0)

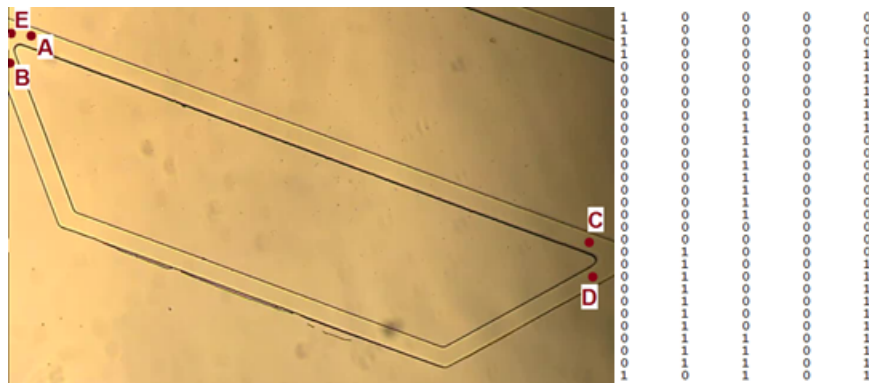


FIGURE 3 – A gauche : position des balises. A droite : exemple de tableau sortie

3.3 Analyse des tableaux de données

On obtient en sortie un tableau à 5 colonnes, pour chaque balise. Le nombre de lignes correspond au nombre d'images qu'on a en sortie du premier programme. A partir de ces tableaux, on peut tracer

le nombre de goutte qui passe sur chaque balise en fonction du temps. Si on compare le tracé pour l'entrée du haut et l'entrée du bas, leur somme devrait donner un tracé identique à celui correspondant à la balise de l'entrée, à un déphasage près. Nous pouvons ainsi voir le « choix » fait par les gouttes à l'entrée de la bifurcation. Nous avons aussi la possibilité d'exploiter ces données autrement : en utilisant le couple de balises entrée-sortie pour chaque branche, on peut remonter au nombre de gouttes dans chaque canal à un instant t . Ce couple forme un système de péage qui permet de suivre les gouttes qui entrent et qui sortent dans chaque branche, ainsi de comptabiliser combien il y en a dans le canal, grâce aux conditions initiales.

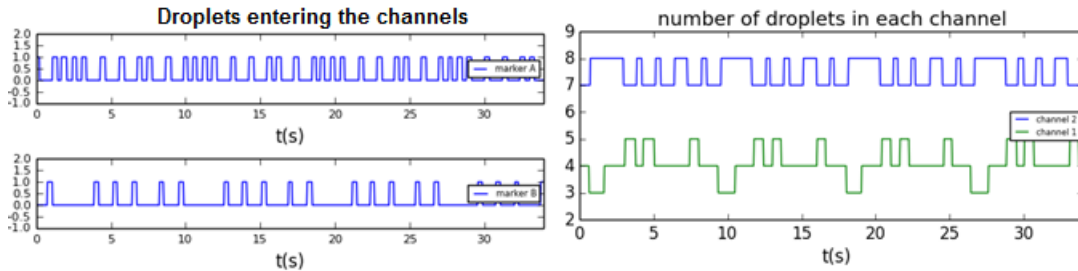


FIGURE 4 – A gauche : valeur des balises pour le canal du haut et du bas en fonction du temps. A droite : nombre de gouttes dans chaque canal en fonction du temps

On s'intéresse ensuite à la transformée de Fourier de ces deux types de signaux périodiques, pour voir si cela donnait des informations sur les propriétés du circuit. Dans ce but, on utilise la fonction "periodogram" de scipy, qui donne la densité spectrale d'un signal échantillonné en passant par la transformée de Fourier directe.

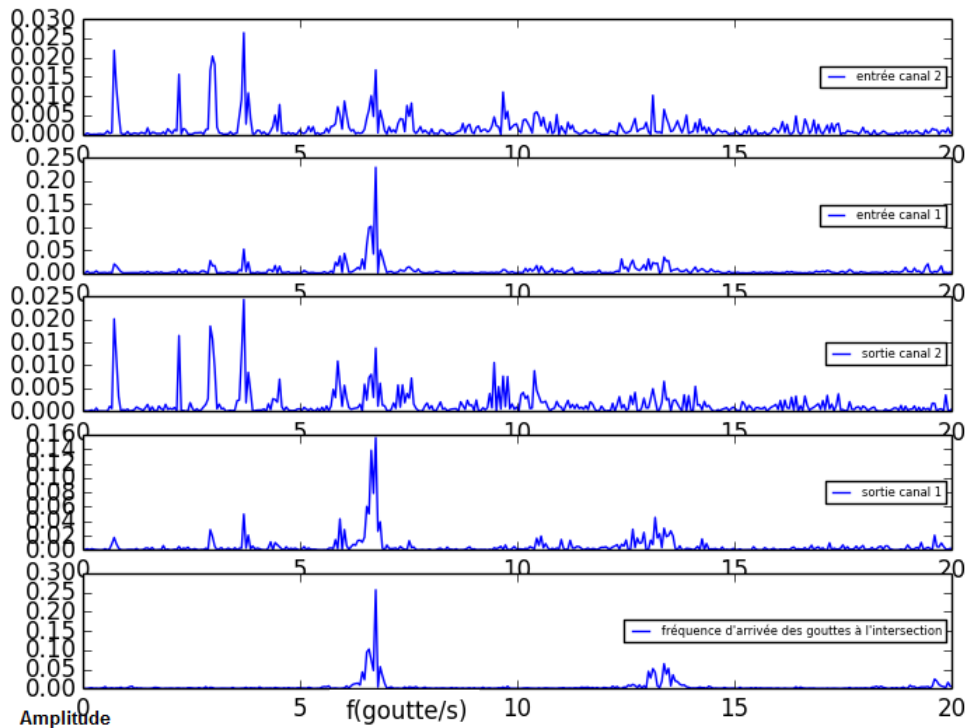


FIGURE 5 – Exemple de densité spectrale d'entropie pour les trains de gouttes à chaque balise

Pour continuer dans cette lancée, on s'intéresse à l'entropie spectrale de nos signaux. Celle-ci est à l'image de l'entropie du système dynamique. Pour cela, on utilise la formule de l'entropie $S \propto k \sum \ln(d)$, car on a une répartition normée de densité.

4 Annexes

4.1 Programme d'extraction des gouttes

```
colorbase = Image("PSE_MARS/Movie01/images/base.jpg")
#image de référence (obtenue lorsque le circuit
#est vierge de gouttes)

colorbase8 = Image(colorbase)
RGBToHLS(colorbase, colorbase8)

h, initial, s = extractChannels(colorbase8)

fond = Image(initial)

for i in range(1, 2500): #Chaque image du film
    j = 3*i #Vu la fréquence d'échantillonnage, 1 image sur 3 suffit
    filename = "PSE_MARS/Movie01/images/goutte"+"%d"%j+".jpg"
    colorim = Image(filename)
    colorim8 = Image(colorim)
    RGBToHLS(colorim, colorim8) # Conversion de la convention de codage de l'image RGB en HLS
    h, encours, s = extractChannels(colorim8) #Extraction des informations sur la saturation
    imin, imfill, imout = Image(encours), Image(encours), Image(encours)
    imfilt = Image(encours)

    sub(initial, encours, imin) #soustraction à l'image de référence

    fillHoles(imin, imfill) #remplissage des gouttes blanches
    ASF(imfill, 7, imout) #filtrage
    hMaxima(imout, 10, imfilt) #repérage des maximums pour obtenir des zones blanches

    write(imfilt, "PSE_MARS/Movie11/images/goutte_c"+"%d"%i+".png")

    compare(imfilt, ">", 0, fond + 1, fond, fond) #superposition des gouttes

write(fond, "PSE_MARS/Movie01/images/fond.png") #lieu de passage des gouttes
```

4.2 Programme d'analyse des trains de gouttes

```
f = open("PSE_MARS/Movie10/tablette.txt", 'w')
l = []
for i in range(1, 990):
    for k in range(1, 6):
        imin = Image("PSE_MARS/Movie10/images/goutte_c"+"%d"%i+".png")
        #Balise k:
        im = Image(imin)
        label = Image(imin)
        balise = Image("PSE_MARS/images/balise_"+"%d"%k+".png")
        compare(balise, "=", 0, 0, imin, im)
        #y a-t-il une goutte sur la balise k?
        H = lambdaLabel(im, 10, label)
        f.write("%d"%H+"\t")

    f.write("\n")

f.close
```

4.3 Traitement des données obtenues

```
from __future__ import division
import matplotlib.pyplot as plt
#import numpy.fft
import scipy.signal
import numpy as np

ini = np.loadtxt("../initial.txt", skiprows = 1) #nombre initial de gouttes
inito = ini[:,0]#dans le canal du haut
iniba = ini[:,1]#dans le canal du bas

g = open("table_sommes.txt", 'w')
som = []

for k in range(9,20):
    plt.close("all")
    l = np.loadtxt("../Movie%d/table.txt"%k, skiprows = 1) #tableau obtenu avec
    #le second programme

    haut = l[:, 0]
    bas = l[:, 1]
    sortiehaut = l[:, 2]
    sortiebas = l[:, 3]
    entre = l[:, 4]

    fin = len(bas)

    sommebas = [int(iniba[k-9])] #tableau nombre de gouttes en bas par image
    sommehaut = [int(inito[k-9])]

    for i in range(1, fin-1):
        if bas[i] - bas[i-1] > 0: #une goutte arrive sur la balise d'entree
            sommebas.append(sommebas[i-1] + 1)
            if sortiebas[i] - sortiebas[i-1] < 0: #une goutte quitte la balise de sortie
                sommebas[i] = sommebas[i] - 1

        else:
            if sortiebas[i] - sortiebas[i-1] < 0: #une goutte sort sans qu'une goutte ne soit entrée
                sommebas.append(sommebas[i-1] - 1)
            else:
                sommebas.append(sommebas[i-1]) #aucune goutte n'entre ou ne sort

        if haut[i] - haut[i-1] > 0: #une goutte entre dans le canal du haut
            sommehaut.append(sommehaut[i-1] + 1)
            if sortiehaut[i] - sortiehaut[i-1] < 0: #une goutte sort du canal du haut
                sommehaut[i] = sommehaut[i] - 1

        else:
            if sortiehaut[i] - sortiehaut[i-1] < 0: #une goutte sort sans qu'une goutte ne soit entrée
                sommehaut.append(sommehaut[i-1] - 1)
            else:
                sommehaut.append(sommehaut[i-1]) #aucune goutte n'entre ou ne sort
```