

# La désinfection par ultraviolets-C

**ROGER Mathilde - 28907**

*Sujet* : **Comment choisir la puissance, la distance et la durée d'éclairage pour assurer une désinfection efficace ?**

Lycée Faidherbe Lille

TIPE 2022 - **SANTÉ - PRÉVENTION**

# Sommaire de la Présentation

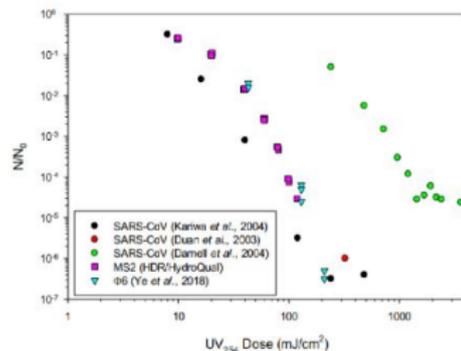
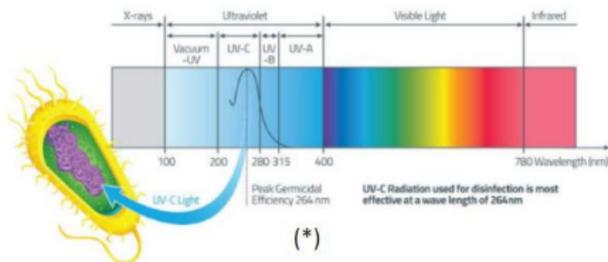
- 1 Introduction
- 2 Modélisation
- 3 Simulation
- 4 Étude expérimentale
- 5 Bilan
- 6 Conclusion
- 7 Annexes

# Désinfecter en utilisant une lumière invisible ?

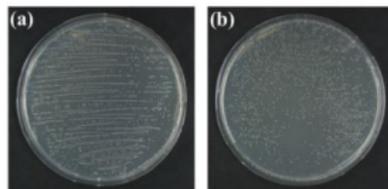
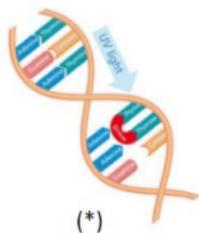


Lampe pour la désinfection UV (source amazon)

# Interaction de l'UV-C avec les virus et bactéries



Courbe de survie UV254nm du COVID dans une suspension aqueuse (\*\*\*)



\*Multiphysics Modeling of UV-C LED Reactor Design for Point-of-Use Drinking Water Disinfection, [https://www.iuva.org/resources/2020\\_Americas\\_Conference](https://www.iuva.org/resources/2020_Americas_Conference)

\*\*source : Total Internal Reflection of Deep-Ultraviolet Light in a Water Waveguide and Its Application to Water Disinfection Technologies, Water 2019, vol 11

\*\*\*SARS-CoV-2 UV Dose-Response Behavior, International Ultraviolet Association, 2020

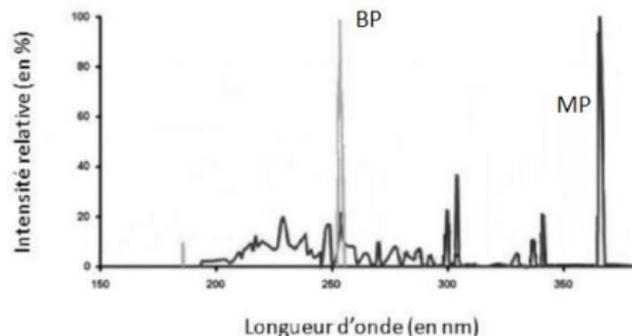
# Les technologies 1: la Lampe à vapeur de mercure



Décharge électrique dans un tube  
Le mercure liquide se vaporise

- imposant
- énergivore

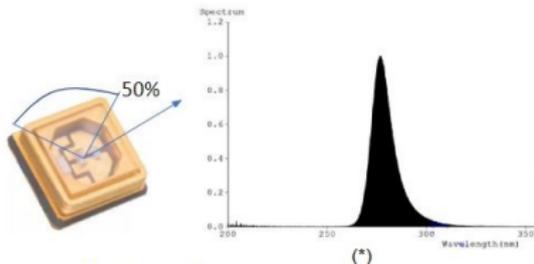
+ Multidirectionnelle 360°  
+ possibilité de choisir la longueur et la puissance  
+ rendement (30 à 40% pour basse pression)



Spectre d'émission d'une lampe à vapeur de mercure basse pression (gris clair) et d'une lampe moyenne pression (noir) \*

(\*) *Hansen K.M.S. et al. (2013). Photolytic removal of DBPs by medium pressure UV in swimming pool water, Sci. Total Environ. ; 443 : 850-856.*

# Les technologies 2: les diodes LED

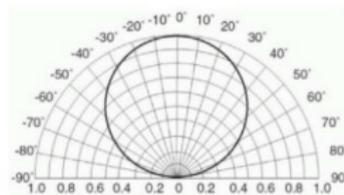
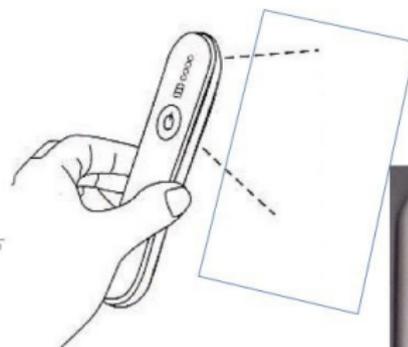


- Point source directionnel
- Moins de 50% d'intensité au-delà de 100 et 120°
- Rendement classique entre 3 et 5%

## Avantages technologiques :

- + Compact et forte densité d'énergie
- + Possibilité de créer des matrices de led
- + Spectre de rayonnement étroit
- + Le rendement évolue sans cesse (auj. jusqu'à 20%)

(\*) LucBei manufacturing store



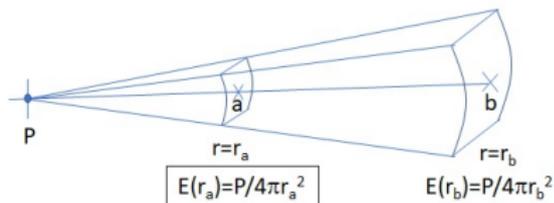
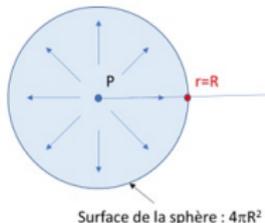
Intensité du rayonnement dans l'espace



Vue dessous

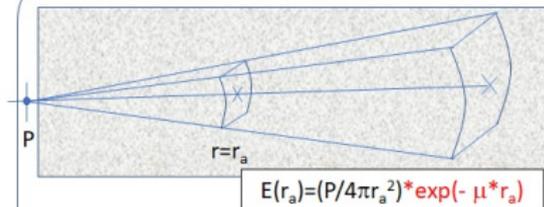
# Modéliser le rayonnement, principes

Point source isotrope de Puissance P  
rayonnant dans toutes les directions



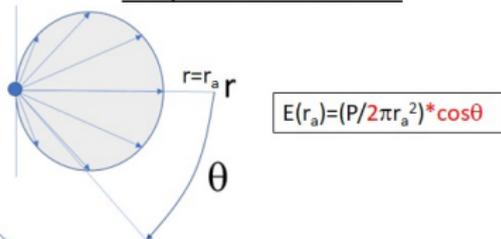
Densité de puissance dans le vide  $E$  (W/cm<sup>2</sup>)

Point source dans un milieu absorbant



Beer Lambert, coefficient d'absorption :  $\mu(\lambda)$

Point source dans un demi espace  
et rayonnement lambertien



# Modélisation : la lampe à vapeur de mercure (1)

$$PW = 6 \text{ W}$$

$$l = 36 \text{ cm}$$

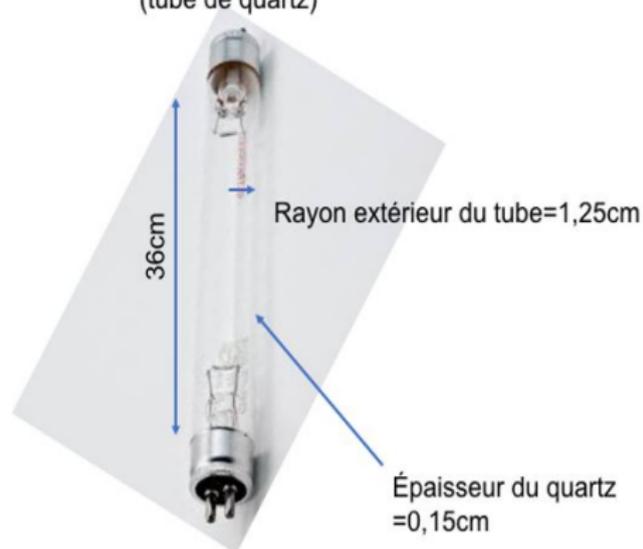
$$\mu_{eau} = 0.431 \text{ cm}^{-1}$$

$$\mu_{quartz} = 0.342 \text{ cm}^{-1}$$

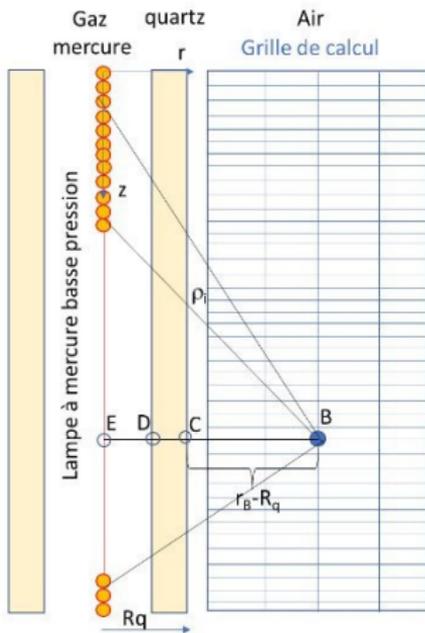
$$r_{ext} = 1.25 \text{ cm}$$

$$e_{tube} = 0.15 \text{ cm}$$

Lampe à mercure basse pression  
(tube de quartz)



# Modélisation : la lampe à vapeur de mercure (2)



## Hypothèses :

La lampe est modélisée par une ligne de 361 points sources isotropes

Le quartz est un milieu absorbant

## Contribution de tous les points sources sur un point de la grille B :

$$\sum_{i=1}^{361} E_{B/i} = \sum_{i=1}^{361} \frac{P}{n * 4\pi * \rho_i^2} \exp\left(-(\mu_{air}(r_B - Rq) + \mu_{quartz}d_{quartz}) \frac{\rho_i}{r_B}\right)$$

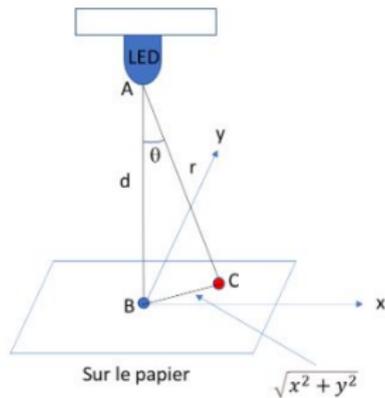
$E_{B/i}$  densité de puissance reçue par B du point source  $i$  (W/cm<sup>2</sup>)

Le quartz et l'air peuvent avoir des facteurs d'atténuation :

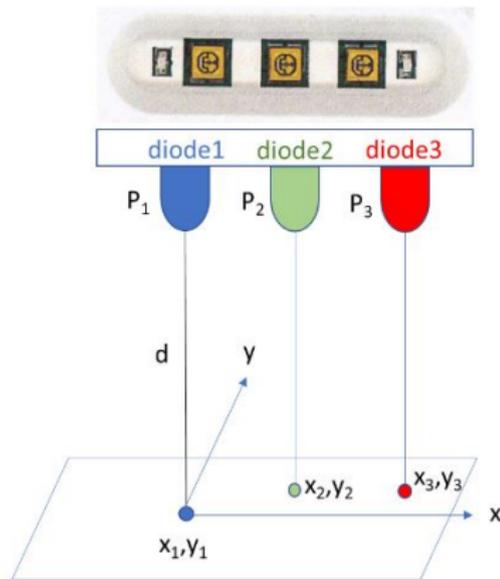
$$\mu_{quartz} = 0,342 \text{ cm}^{-1}$$

$\mu_{air} = 0 \text{ cm}^{-1}$ , l'air n'absorbe pas le rayonnement UV

# Modélisation : les LED



$$E(r, \theta) = \frac{P \cos \theta}{\pi r^2} = \frac{P}{\pi} * \frac{d}{\sqrt{d^2 + x^2 + y^2}} * \frac{1}{d^2 + (x^2 + y^2)}$$



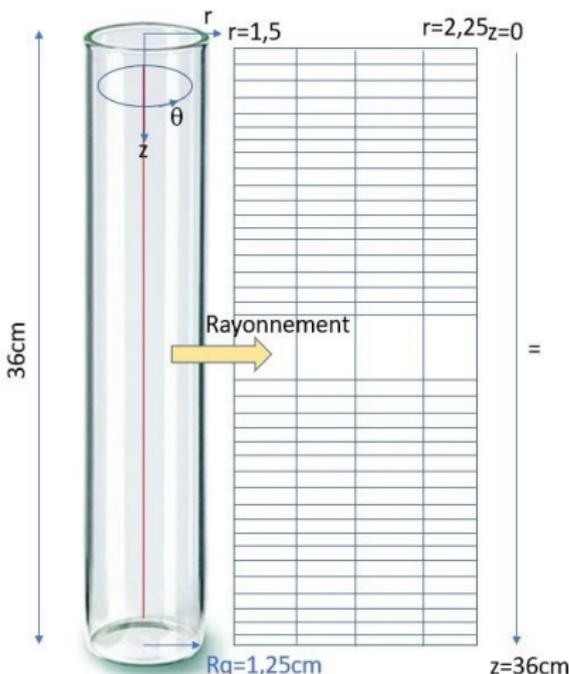
$$E(d, x, y) = \sum_{i=1}^3 \frac{P_i d}{\pi (d^2 + (x - x_i)^2 + (y - y_i)^2)^{\frac{3}{2}}}$$

# Simulation : la lampe à vapeur de mercure

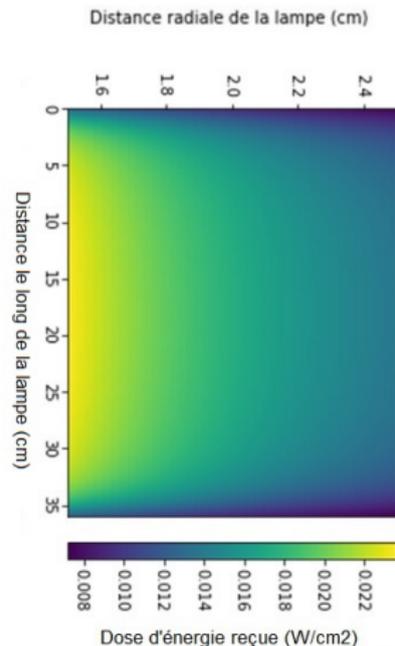


Lampe UV

||

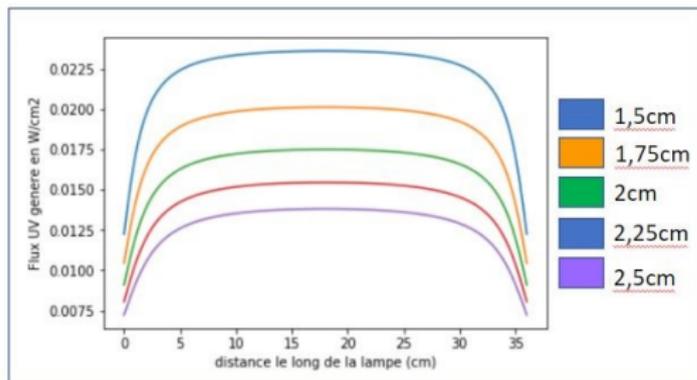


Le rayonnement est calculé sur une grille 2D  $(r,z)$   
Il ne dépend pas de l'angle  $\theta$  (Invariance)

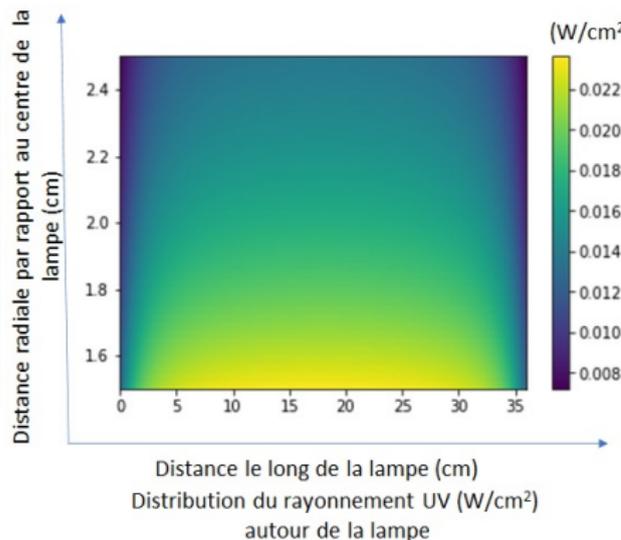


Le champ de rayonnement est interpolé sur la surface

# Distribution spatiale du rayonnement

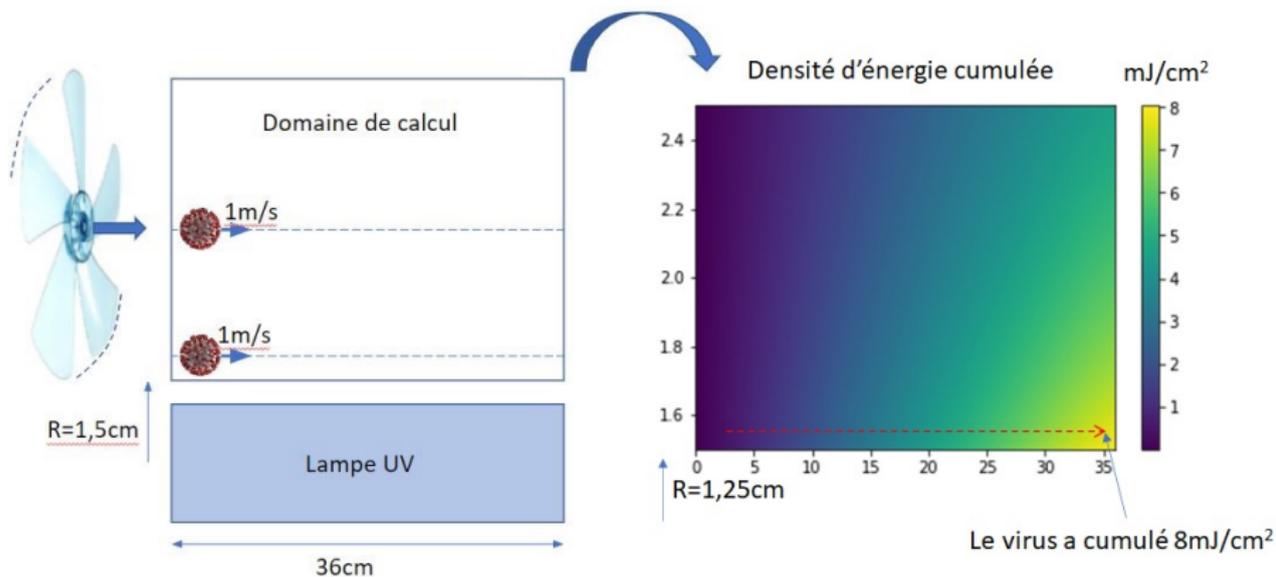


Distribution du rayonnement UV ( $W/cm^2$ ) le long de la lampe pour différentes distances radiales



Distribution du rayonnement UV ( $W/cm^2$ ) autour de la lampe

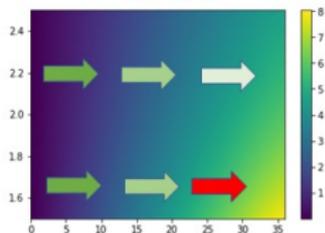
# Densité d'énergie absorbée par les virus



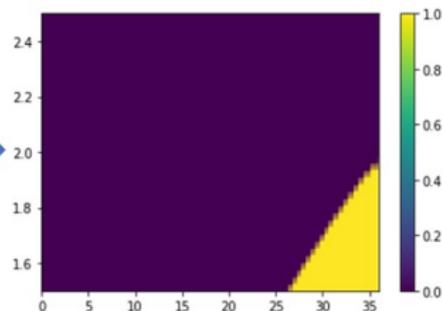
Dose cumulée de gauche à droite en  $\text{mJ/cm}^2$  par un virus se déplaçant suivant  $z$  (parallèlement au tube) à  $1\text{m/s}$  de gauche à droite en fonction de la distance au centre du tube de quartz

# Détermination d'une zone de mortalité

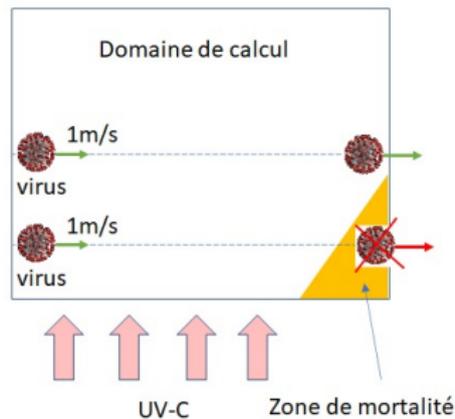
Densité d'énergie cumulée  
(mJ/cm<sup>2</sup>)



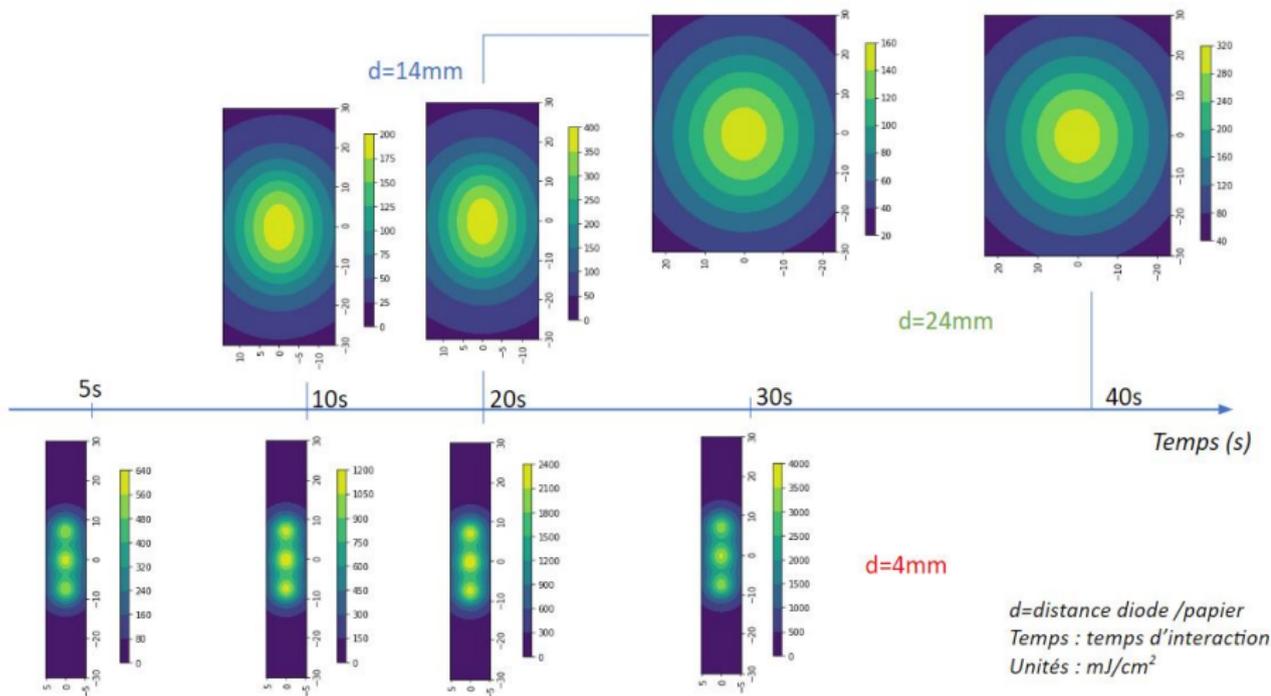
Carte de mortalité pour  
 $v=1\text{m/s}$  et seuil =  $6\text{mJ/cm}^2$



0 = virus vivant / 1 = virus détruit  
On fixe un seuil de mortalité à  $6\text{mJ/cm}^2$   
pour un virus donné



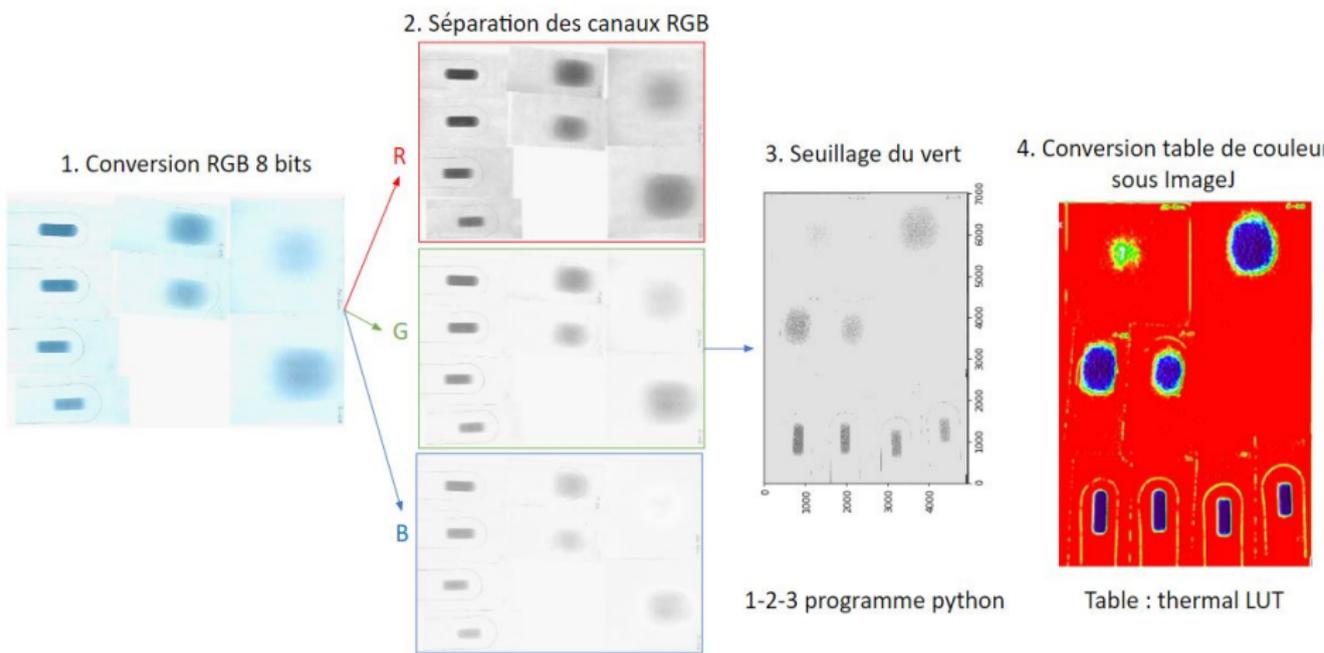
# Simulation : les leds



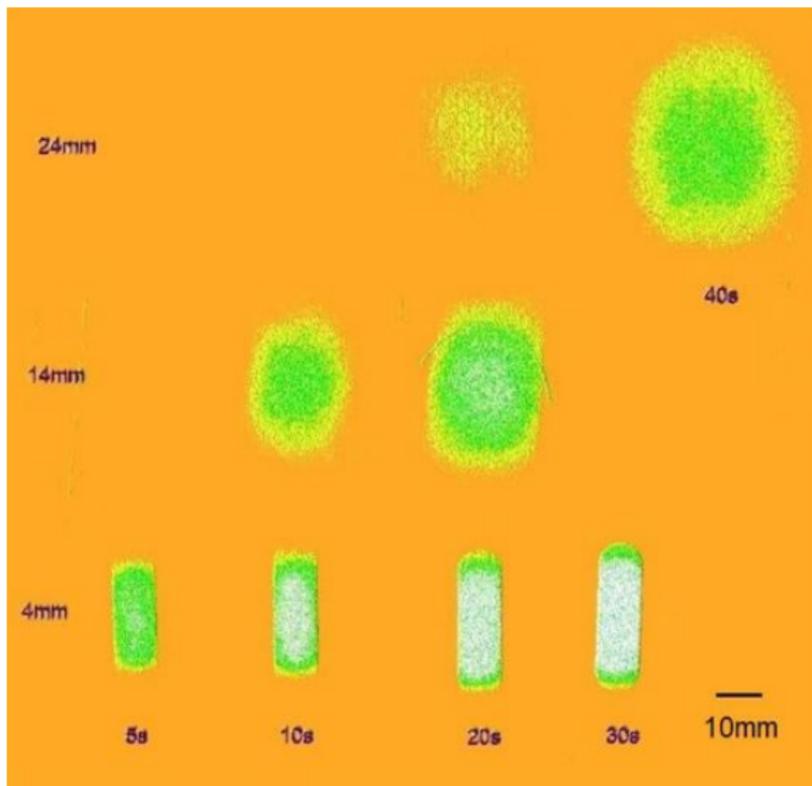
# Proposition d'une méthode d'identification de la distribution du rayonnement UV-C



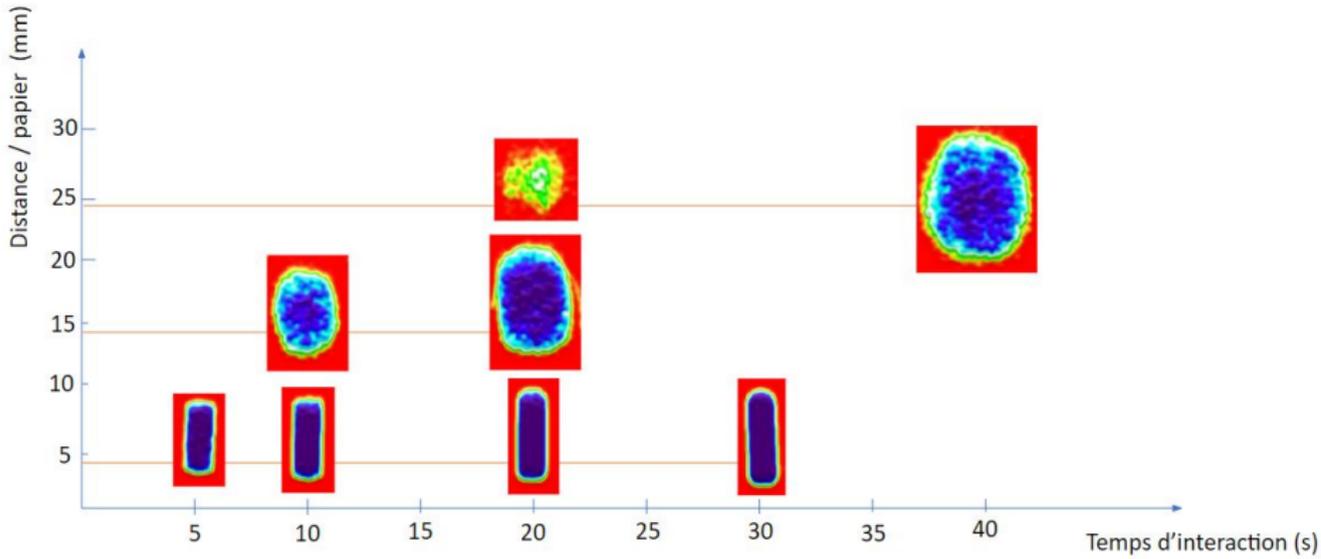
# Traitement d'images des cyanotypes



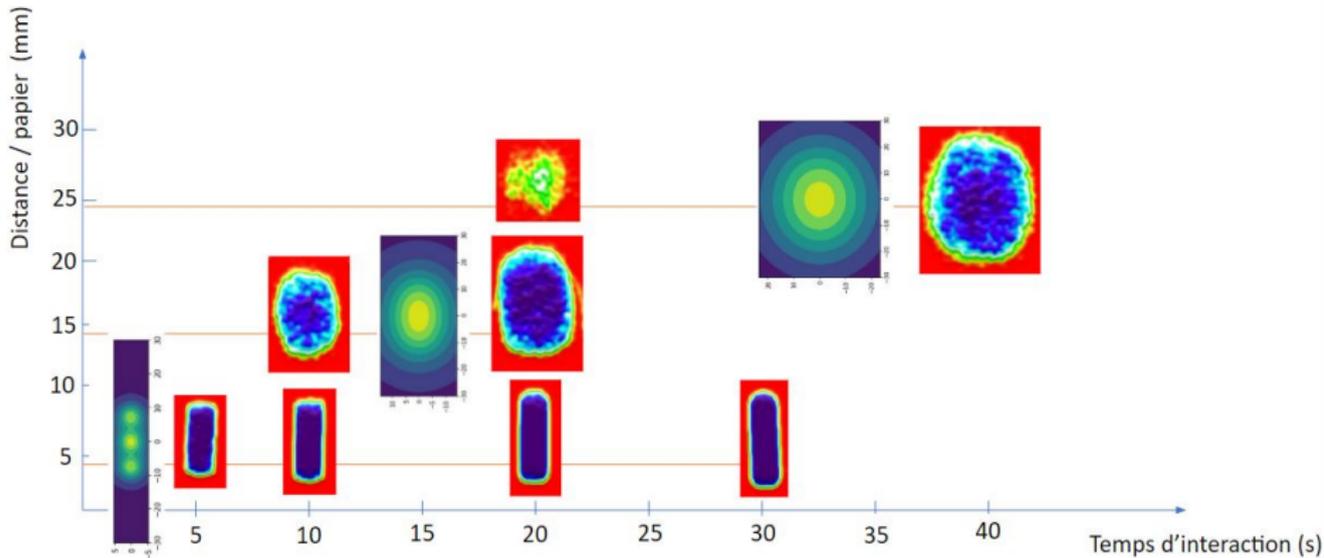
# traitement d'images du canal rouge sous ImageJ



# Classification des cyanotypes

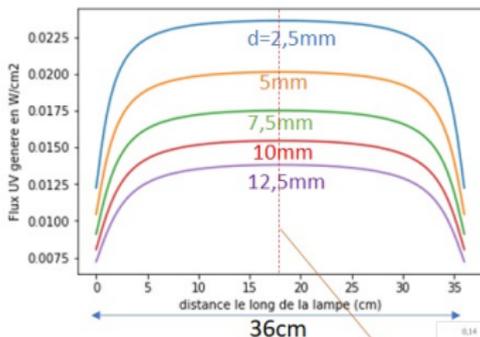


# Comparaison cyanotypes/simulations LED

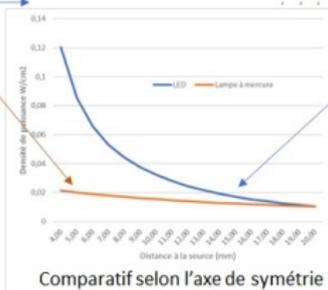
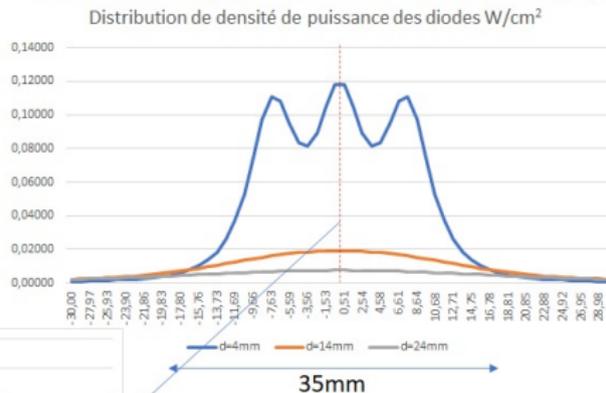


# Comparaison technologie LED et Lampe à vapeur de Mercure

Lampe 6W / longueur 36cm (0,16W/cm de tube)



3 leds de 50mW (0,042W/cm selon l'axe de la lampe)

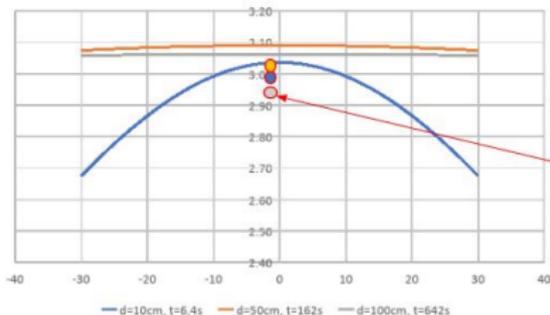


# Validation du modèle LED

Société Stanley Electric (<https://www.youtube.com/watch?v=FFMZNMIsAXw>)

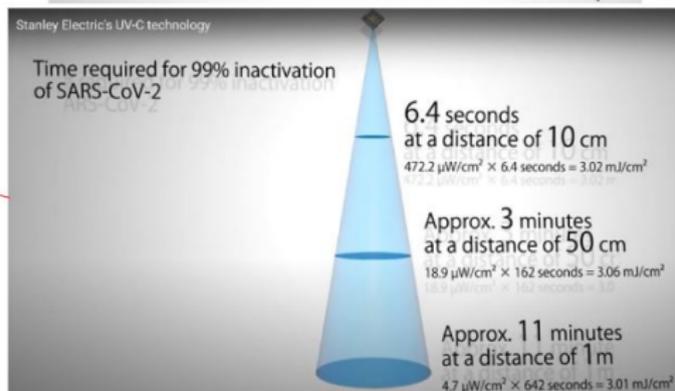
3 led UV-C de 50mW espacées de 10, 50 et 100cm de la zone de désinfection

Résultats de la simulation python  
Distribution de densité d'énergie mJ/cm<sup>2</sup>



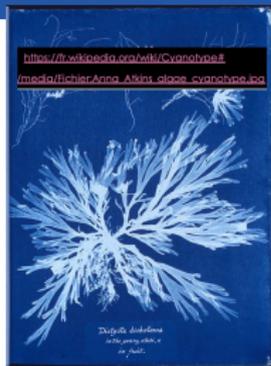
distance, temps	Energie cumulée (mJ/cm <sup>2</sup> )	
	Python	stanley
10cm et 6,4s	3,04	3,02
50cm et 162s	3,09	3,06
100cm et 642s	3,06	3,01

Irradiation simulation of 265 nm/50 mW LED × 3 pcs



Simulation de la société STANLEY Electric : 3 leds de 50mW

# Conclusion et perspectives



- Simuler le rayonnement UV-C permet de vérifier l'efficacité des technologies de désinfection
- Les cyanotypes fournissent une carte précise du rayonnement UV-C
- Les LED supplanteront les lampes à vapeur de mercure car leur rendement augmente sans cesse
- Les LED pourraient permettre de désinfecter des habitacles intérieurs de voitures

# Interview de Talib Dbouk

## Pouvez vous vous présenter en quelques phrases ?

Je suis enseignant-chercheur (Associate Professor) au sein de Faculté de Médecine de l'Université de Nicosia à Chypre. En parallèle je suis également le directeur de recherche de l'institut de la défense et la sécurité (DSRI) dans la même Université (<https://www.unic.ac.cy/dbouk-talib/>). Je suis également un enseignant-chercheur de l'Institut Mines-Telecom.

Mes thématiques et travaux de recherches couvrent un aspect large de différents domaines, qui sont centralisés autour de la modélisation et simulation des phénomènes complexes multi-physiques, comme: l'optimisation, la mécanique des fluides, le transfert de chaleur, la rhéologie, la tribologie, l'aérodynamique, l'aéroacoustique, l'intelligence artificielle...

## En quoi consiste votre activité de chercheur sur le COVID et la simulation ?

Les expérimentations (e.g. manips de mesure) dans ce domaine (COVID) sont difficiles et à très haut risque (contamination, échelle, etc), mes activités de recherche dans les derniers 16 mois ont eu l'objectif de modéliser et simuler le transport de virus dans l'air (en modes intérieur et/ou extérieur). Depuis le début de la pandémie en Mars 2020, je cherche à mieux comprendre les mécanismes responsable de la transmission de virus (e.g. effet des conditions de l'environnement comme la vitesse du vent, température, l'humidité et récemment le Pollen).

## D'après vous quelle est la vraie distance sociale à respecter pour se protéger du COVID 19 ?

La vraie distance sociale à respecter, dépendrait toujours des conditions de l'environnement. Par exemple, si nous sommes à l'intérieur et qu'il n'y a pas de convection naturelle importante, une distance de 2 mètres est suffisante. Par contre s'il existe une vitesse de l'air (comme une convection importante intérieure ou extérieure par le vent) la distance sociale augmente. Dans une publication nous avons montré qu'un nuage de gouttelettes de salive peut aller jusqu'à 6 mètres en présence de vent à une vitesse de 15 km/h suite a une toux d'une personne infectée.

## Dans votre activité, vous utilisez beaucoup la simulation avec des gros modèles numériques, est ce que c'est une démarche classique pour un chercheur ?

Ce n'est pas une démarche classique car nous n'avons pas simplement utilisé des modèles. Nous avons développé des modèles physiques avancés afin de mieux enrichir la physique responsable de transport de particules de virus dans l'air (i.e. phénomènes de l'évaporation, filtration via un masque, dynamique, etc). Bien sur à noter que chaque méthode et modèle numérique à un certain niveau de précision. Enfin, c'est un compromis entre le temps de calcul et la précision souhaitée/acceptée.

# Interview de Talib Dbouk

## Quels sont les facteurs environnementaux qui favorisent la propagation du COVID ?

Nous avons trouvé que les facteurs environnementaux qui favorisent la propagation du COVID sont des combinaisons entre 3 paramètres: la température, la vitesse du vent et l'humidité. Dans un autre papier, on peut clairement voir cette relation entre ces 3 paramètres dans la figure 5 jointe ou on montre la variation du coefficient de transmission en fonction de la température  $T$ , vitesse du vent  $U$  et l'humidité relative  $RH$ .

## Que pensez vous d'utiliser l'ultraviolet pour désinfecter l'air ?

C'est une technologie appropriée et émergente mais elle nécessiterait des développements de recherche afin de bien l'adapter aux applications de désinfection de l'air.

## Est ce possible d'après vous de désinfecter en continu des pièces d'habitation de façon efficace ?

Malheureusement, je ne peux pas vous répondre à cette question. C'est une question à laquelle je cherche moi-même à répondre via mes travaux de recherche et développements actuels.

# Comparaison cyanotype / simulation

Distance diode papier : 4mm, temps d'interaction 5s

Cyanotype original

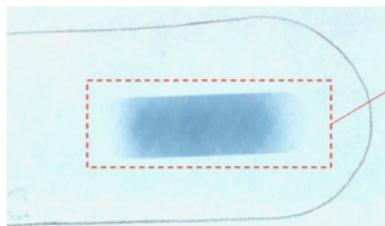


Image Red en RGB

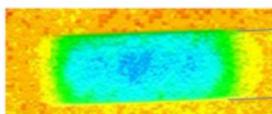


Table spectrum LUT inversée

Image Green en RGB

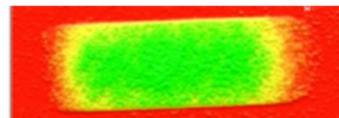
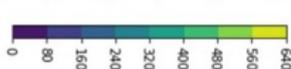
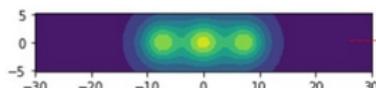


Table thermal LUT

1cm

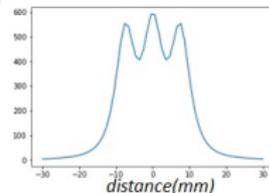
Simulation sous python :

densité d'énergie cumulée en 5s



mJ/cm2

mJ/cm2



# Code lampe à vapeur de mercure 1

```
import numpy as np
import matplotlib.pyplot as plt

# CONSTANTES
PW=6; #Puissance totale de la lampe en Watts
LL=36; #longueur de la lampe UV
AE=0.431; #coefficient d'Attenuation de l'eau (par cm)
AE=0; # l air remplace l eau donc attenuation est remplace par 0
AQ=0.342; #coefficient d'Attenuation du quartz (par cm)
RA=2.25; #Rayon pour le postraitement comparatif
RQ=1.25; #Rayon exterieur du tube de quartz
EQ=0.15; # epaisseur du tube de quartz
NIZ=360; #nombre d'intervalles le long de la lampe

NP=NIZ+1; #nombre de points rayonnants sur la lampe
NR=20; #Nombre de rayons
VR=np.linspace(1.35,3.25,NR) #tableau de la liste des rayons

DZ=0.1; #pas de découpage de l espace selon selon z (cm)
VZ=np.linspace(0,LL,num=NP); #discretisation de la lampe en 361 points source

VP=100; #vitesse des particules ou virus (ils se déplacent selon z) 100cm/s
DT=DZ/VP; #pas de temps en seconde lorsque le virus saute d'un point à l'autre
```

# Code lampe à vapeur de mercure 2

```
#fonctions
def UVC(PW,RQ,EQ,AE,AQ,VR,VZ):
    #UVC unités W/cm2
    #PW puissance totale (W),RQ rayon externe quartz (cm),EQ epaisseur quartz (cm)
    #AE coeff atténuation air (cm-1), AQ coeff atténuation quartz (cm-1)
    #VR tableau numpy des coordonnées de la grille selon r,
    #VZ tableau numpy des coordonnées de la grille selon z et des points sources selon z
    NR=np.size(VR)
    NP=np.size(VZ)

    IR=np.zeros((NR, NP)); #on initialise la grille des points ou on va calculer le champ UV

    #attention le premier indice de IR est zero, l indice du tableau commence a zero
    #par exemple IR[0,5] est le champ UV au point 0,5 premier point suivant R et 6eme suivant Z

    IRP=PW/(NP*4*np.pi); #puissance par point divise par 4 pi (pi=np.pi) angle solide = 4pi radians

    for k in range(NP):
        for i in range(NR):
            for j in range(NP):
                old=IR[i,j]
                rho2=(VZ[k]-VZ[j])**2+(VR[i])**2
                rho=np.sqrt(rho2)
                ATT=-(AE*(VR[i]-RQ)+AQ*EQ)
                IR[i,j]=old+(IRP/rho2)*np.exp(ATT*(rho/VR[i]))
    return IR
# Fin des fonctions
#lancement de la fonction de calcul du champ UVC pour la lampe
IR=UVC(PW,RQ,EQ,AE,AQ,VR,VZ)
```

# Code lampe à vapeur de mercure 3

```
#tracé du champ UV pour le premier rayon de la grille
fig=plt.figure()
ax=fig.add_subplot(111)
ax.plot(VZ,IR[0,:])
plt.xlabel('distance le long de la lampe (cm)')
plt.ylabel('Flux UV genere en W/cm2')
plt.show()

#tracé du champ UV pour le second rayon de la grille

fig=plt.figure()
ax=fig.add_subplot(111)
ax.plot(VZ,IR[1,:])
plt.xlabel('distance le long de la lampe (cm)')
plt.ylabel('Flux UV genere en W/cm2')
plt.show()

#tracé du champ UV pour les 1er et second rayons

fig=plt.figure()
ax=fig.add_subplot(111)
ax.plot(VZ,IR[0,:])
ax.plot(VZ,IR[1,:])
plt.xlabel('distance le long de la lampe (cm)')
plt.ylabel('Flux UV genere en W/cm2')
plt.show()

#tracé du champ UV pour tous les rayons

fig=plt.figure()
ax=fig.add_subplot(111)
for l in range(NR):
    ax.plot(VZ,IR[l,:])
plt.xlabel('distance le long de la lampe (cm)')
plt.ylabel('Flux UV genere en W/cm2')
plt.show()
```

# Code lampe à vapeur de mercure 4

```
#On multiplie par le temps pour calculer une énergie pour un objet fixe
#placé pendant un temps tps a proximité et on convertit en J/mm2
tps=10.
IR2=(tps/100.)*IR
fig=plt.figure()
ax=fig.add_subplot(111)
for l in range(NR):
    ax.plot(VZ,IR2[l,:])
plt.xlabel('distance le long de la lampe (cm)')
plt.ylabel('Energie UV cumulee en J/mm2')
plt.show()

#On multiplie par le temps pour calculer une énergie pour un objet fixe
#placé pendant un temps tps a proximité et on convertit en J/cm2
tps=20.
IR2=tps*IR
fig=plt.figure()
ax=fig.add_subplot(111)
for l in range(NR):
    ax.plot(VZ,IR2[l,:])
plt.xlabel('distance le long de la lampe (cm)')
plt.ylabel('Energie UV cumulee en J/cm2')
plt.show()
```

# Code lampe à vapeur de mercure 5

```
#Calcul des doses recues par des virus se déplaçant
#parallelement à la lampe (axe z) à des vitesses fixes de 1 m/s
#mais pour des rayons différents donc des distances plus ou moins
#proches du tube

#on prend les mêmes rayons R=1.5cm à R=2.5cm
#comme le tube de quartz à un rayon de 1.25cm, il faut le retrancher
#pour avoir la distance au tube

#Le tableau DOSE[i] stocke la dose UV cumulée pendant la traversée
#comme la vitesse est constante et les virus se déplacent sur la grille selon z
#le pas de temps entre point de la grille est constant et vaut DT
#le calcul du cumul est facile, c'est le produit DT par la somme des valeurs de la ligne

DOSE=np.zeros(NR)
DIST=VR-RQ

for i in range(NR):
    DOSE[i]=1000*DT*np.sum(IR[i,:])

#DOSE en mJ/cm2 donc facteur 1000 pour conversion Joule en milliJoules
```

# Code lampe à vapeur de mercure 6

```
#tracé de la dose reçu par un virus se déplaçant suivant z à 1m/s parallèlement au tube  
#en fonction de sa distance à l'extérieur du tube de quartz  
fig=plt.figure()  
ax=fig.add_subplot(111)  
ax.plot(DIST,DOSE)  
plt.xlabel("Distance le long de la rampe (cm)")  
plt.ylabel('Dose reçue en mJ/cm2')  
plt.show()  
  
#Trace de la carte de rayonnement autour de la lampe UV (W/cm2)  
  
X,Y=np.meshgrid(VZ,VR)  
plt.pcolormesh(X,Y,IR,shading="gouraud")  
plt.colorbar(label="Dose d'énergie reçue (W/cm^2)")  
plt.xlabel("Distance le long de la rampe (cm)")  
plt.ylabel('Distance radiale de la lampe (cm)')  
plt.show()
```

# Code lampe à vapeur de mercure 7

```
#Trace de la carte de rayonnement autour de la lampe UV et trajectoire une particule

#distance (cm) à la lampe où l'on veut observer la trajectoire du virus
dist=2

#dose cumulee mortelle pour le virus mJ/cm2
dosecrit=6

DEB=np.linspace(0,10,11)
DEBZ=2*np.ones(11)
FIN=np.linspace(12,36,25)
FINZ=2*np.ones(25)
X,Y=np.meshgrid(VZ,VR)
fig, ax = plt.subplots()
ax.pcolormesh(X,Y,IR,shading="gouraud")
ax.plot(DEB, DEBZ, 'o', color='g')
ax.plot(FIN, FINZ, 'o', color='r')
plt.show()

#Matrice IRC qui cumule la puissance rayonnee colonne par colonne
IRC=IR
for j in range(1,NP):
    IRC[:,j]=IRC[:,j-1]+IR[:,j]

X,Y=np.meshgrid(VZ,VR)
plt.pcolormesh(X,Y,IRC,shading="gouraud")
plt.colorbar()
plt.show()
```

# Code lampe à vapeur de mercure 8

```

#Matrice ERC qui cumule l'énergie colonne par colonne mJ/cm2
print(DT)
ERC=DT*IRC*1000
X,Y=np.meshgrid(VZ,VR)
plt.pcolormesh(X,Y,ERC,shading="gouraud")
plt.colorbar(label="Dose d'énergie reçu (mJ/cm^2)")
plt.xlabel("Distance le long de la rampe (cm)")
plt.ylabel("Distance radiale de la lampe (cm)")
plt.show()

#SEUIL d'énergie cumulée pour tuer le virus

SEUIL=6 # mJ
ERC2=ERC

for i in range(NR):
    for j in range(NP):
        if ERC2[i,j]<SEUIL:
            ERC2[i,j]=0
        else:
            ERC2[i,j]=1

X,Y=np.meshgrid(VZ,VR)
plt.pcolormesh(X,Y,ERC2,shading="gouraud")
plt.colorbar(label="Zone Mortelle ou Non")
plt.xlabel("Distance le long de la rampe (cm)")
plt.ylabel("Distance radiale de la lampe (cm)")
plt.show()

****tracé de la densité de puissance W/cm2 en fonction de la distance en mm au centre de la lampe
#il faut enlever le rayon du quart a VR pour demarrer à zéro contre le quartz et convertir VR de cm a mm
VR2=VR-1.25
VR2=VR2*10.
plt.plot(VR2,IR[:,180])
np.savetxt("liste_dmercure.csv", VR2, delimiter=",")
np.savetxt("densite_mercure.csv", IR[:,180], delimiter=",")

```

# Code LED 1

```
import numpy as np
import matplotlib.pyplot as plt

# CONSTANTES

P=0.05 #puissance de la led (W)
d=4 #distance led papier (mm)
tps=20 #temps d'allumage des leds en s

x1=-7.5
y1=0
x2=0
y2=0
x3=7.5
y3=0

ymin=-0.90886*d-1.51945
ymax=0.90886*d+1.51945

x = np.linspace(-30, 30, 60)
y = np.linspace(ymin, ymax, 40)
```

# Code LED 2

```
# ***fonctions

#Puissance en W par unité de surface mm2
def puissance(P,d,x1,y1,x2,y2,x3,y3,x,y):
    X, Y = np.meshgrid(x, y)
    Z1=((P*d)/(np.pi*(d**2+(X-x1)**2+(Y-y1)**2)**(3./2)))
    Z2=((P*d)/(np.pi*(d**2+(X-x2)**2+(Y-y2)**2)**(3./2)))
    Z3=((P*d)/(np.pi*(d**2+(X-x3)**2+(Y-y3)**2)**(3./2)))
    Z=Z1+Z2+Z3
    return Z

#energie J par unité de surface mm2
def energie(P,d,x1,y1,x2,y2,x3,y3,x,y,tps):
    X, Y = np.meshgrid(x, y)
    E1=((P*d*tps)/(np.pi*(d**2+(X-x1)**2+(Y-y1)**2)**(3./2)))
    E2=((P*d*tps)/(np.pi*(d**2+(X-x2)**2+(Y-y2)**2)**(3./2)))
    E3=((P*d*tps)/(np.pi*(d**2+(X-x3)**2+(Y-y3)**2)**(3./2)))
    E=E1+E2+E3
    return E

#densité de Puissance W/mm2 en fonction de la distance d du papier au centre x=y=0 des leds
def Pd(P,ld,x1,y1,x2,y2,x3,y3):
    Pd1=((P*ld)/(np.pi*(ld**2+(0-x1)**2+(0-y1)**2)**(3./2)))
    Pd2=((P*ld)/(np.pi*(ld**2+(0-x2)**2+(0-y2)**2)**(3./2)))
    Pd3=((P*ld)/(np.pi*(ld**2+(0-x3)**2+(0-y3)**2)**(3./2)))
    Pt=Pd1+Pd2+Pd3
    return Pt

# ***fin des fonctions
```

# Code LED 3

```
# calcul du tableau des valeurs de Z
Z=puissance(P,d,x1,y1,x2,y2,x3,y3,x,y)

#Puissance en W par unité de surface cm2
h = plt.contourf(x, y, Z)
plt.axis('scaled')
plt.colorbar()
plt.show()

#energie mJ par unité de surface cm2
Z2=Z*tps*1000.
#on aurait pu ecrire Z2=1000.*energie(P,d,x1,y1,x2,y2,x3,y3,x,y,tps)
h = plt.contourf(x, y, Z2)
plt.axis('scaled')
plt.colorbar()
plt.show()
```

# Code LED 4

```
#energie mJ par unité de surface cm2
Z2=Z*tps*1000.
#on aurait pu écrire Z2=1000.*energie(P,d,x1,y1,x2,y2,x3,y3,x,y,tps)
h = plt.contourf(x, y, Z2)
plt.axis('scaled')
plt.colorbar()
plt.show()

#tracé selon x de la distribution de densité de puissance W/cm2
#Z[19] vecteur sur la ligne de symétrie du milieu y=0, 19 car 40 valeurs en y et commence à zéro,
# 19=20eme valeur sur 40
plt.plot(x,Z[19])

#trace selon x de la distribution de densité d'énergie mJ/cm2
#Z[19] vecteur sur la ligne de symétrie du milieu y=0, 19 car 40 valeurs en y et commence à zéro,
# 19=20eme valeur sur 40
plt.plot(x,Z2[19])

#enregistrer des vecteurs au format excel pour traiter les courbes sous excel

np.savetxt("x.csv", x, delimiter=",")

np.savetxt("z4_5.csv", Z[19], delimiter=",")

#tracé de la densité de puissance W/cm2 au lieu de W/mm2 en fct distance face a x=y=0
#liste des distances pour le calcul en mm
liste_d=np.linspace(1,20,20)
densite=100*Pd(P,liste_d,x1,y1,x2,y2,x3,y3)
plt.plot(liste_d,densite)

np.savetxt("liste_d.csv", liste_d, delimiter=",")
np.savetxt("densite_led.csv", densite, delimiter=",")
```

# Code traitement d'images

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from PIL import ImageFilter

img=Image.open("cyanotype.tif")
rgbimg = img.convert('RGB')
red,green,blue=rgbimg.split()

image_array1 = np.array( green )
plt.imshow( image_array1 )
plt.show()
plt.hist(image_array1.flatten(), bins = 20, density = True , alpha = .5 , edgecolor = 'black', color = 'red')
plt.show()

i,j=green.size
seuil=225
#tous les pixels superieurs a seuil seront remplaces par la valeur seuil
for x in range(i):
    for y in range(j):
        pixel=green.getpixel((x,y))
        if pixel>seuil:
            green.putpixel((x,y),seuil)

plt.imshow( green )
plt.show()

image_array2 = np.array( green )
plt.imshow( image_array2 )
plt.show()
plt.hist(image_array2.flatten(), bins = 20, density = True , alpha = .5 , edgecolor = 'black', color = 'red')
plt.show()
green.show()

#sauvegarder l'image verte filteree en green.jpg
im1 = green.save("green.jpg")
```