TP3_IMA 203 _ BACCARI MOHAMED MOOTEZ

Question 1.1 :

- Dans la fonction « minimisation quadratique » on initialise le vecteur K avec deux filtres « Kx » et « Ky » permettant un calcul de gradient horizontale et verticale et le filtre « delta ». On ajoute dans le vecteur V deux matrices nulles ainsi que l'image « v ».
- Grace à ces deux vecteurs, on appelle L'outil « resoud_quad_fourier » qui permet de trouver une image qui minimise sum_i || K_i conv im - V_i||^2 où les K_i et les Vi sont des filtres et des images respectivement.

Question 1.2 :

- Lors de l'application de la restauration quadratique, on remarque que pour des faibles valeurs de lambda le programme ne modifie pas l'image.
  Par contre, pour le cas d'une très grande valeur de lambda, on remarque que l'image devient floue.

Figure 1: RESTQUAD_LAMB=0,01

Figure 2:RESTQUAD_LAMB=0,01

Question 1.3 :

- En utilisant la méthode de dichotomie, on essaye d'estimer le paramètre "lambda" pour lequel norm2(u_tilde - v) ~ norm2(u-v) grâce à l'algorithme suivant:

```python
#%% Question 1.3:
imb=degrade_image(im,5)
objectif=norm2(imb-im)
lambdamin=0.001
lambdamax=5
resmin=minimisation_quadratique(imb,lambdamin)
resmax=minimisation_quadratique(imb,lambdamax)
Emin=norm2(imb-resmin)
Emax=norm2(imb-resmax)
for k in range (20):
    approx=(lambdamin+lambdamax)/2
    resapprox=minimisation_quadratique(imb,approx)
    Eapprox=norm2(imb-resapprox)
    if Eapprox> objectif :
        lambdamax=approx
        Emax=Eapprox
        resmax=resapprox
    else:
        lambdamin=approx
        Emin=Eapprox
        resmin=resapprox
print(approx)
```

- On trouve que la meilleure valeur de lambda proche de la norme de l'attache aux données est égale à "approx=0,3311"

Question 1.4 :

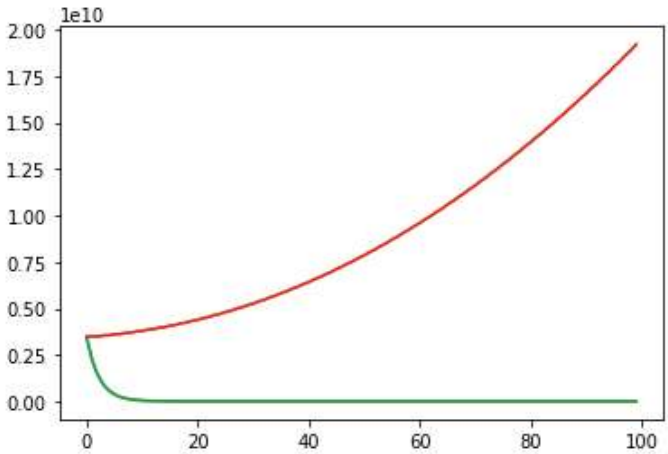
- Pour trouver le paramètre "lambda" qui minimise norm2(u_tilde - u), on applique l'algorithme suivant :

```python
#%% Question 1.4:
erreur=[]
lambdas=[]
vk=np.arange(-1,0,0.01);#on prend des valeur entre 0,1 et 1 vu qu'on sait déja que le lambda
#qu'on cherche doit être inférieur à "lambda =0,3311"
for k in vk:
    print (k)
    restq=minimisation_quadratique(imb,10**(k))
    lambdas.append(10**(k))
    erreur.append(norm2(im-restq))
#print(lambdas)
#print(erreur)
print(min(erreur))
print(erreur.index(min(erreur)))
print(lambdas[erreur.index(min(erreur))])
```

- On trouve que la meilleure valeur de lambda qui minimise norm2(u_tilde - u) est égale à 0.117. Ceci est logique vu que cette valeur est inférieure à "approx=0.3311"

Question 2.1 :

```
#%% Question 2.1:
myim=imread('lena.tif')
imb=degrade_image(myim,25)
(u,energ)=minimise_TV_gradient(imb,1,0.1,100)    # pas = 0.1
(u,energ2)=minimise_TV_gradient(imb,1,1,100)       # pas = 1
plt.plot(energ)
plt.plot(energ2)
```



- En exécutant le programme qui renvoie l'évolution de l'énergie, on remarque que l'utilisation de la méthode de descente de gradient à « pas constant » donne des mauvais résultats. En effet, pour différentes valeurs du « pas », on n'a pas toujours le même résultat.

Question 2.2 :

```
#%% Question 2.2 :
(u,energ)=minimise_TV_gradient(imb,40,0.1,500)
grad=E2_nonperiodique(u,imb,40)
uchamb=vartotale_Chambolle(imb,40,itmax=30)
chamb=E2_nonperiodique(uchamb,imb,40)
print(chamb/grad)
```

- En utilisant le programme « vartotale_Chambolle » au même problème posé par E2, on remarque que même si on applique plus que 500 fois la méthode du gradient avec un pas 0,01, on n'arrive pas à des résultats équivalents par les deux méthodes.

⇨ La méthode de Chambolle est beaucoup plus rapide.

Question 3 :

```
#%% Question 3
im=imread('lena.tif')
imb=degrade_image(im,25)

# d'abord le lambda optimal pour la partie quadratique.
vk=np.arange(-1,1,0.05)
errbest=norm2(imb)+10
for k in vk:
    lamb=10**k
    resquad=minimisation_quadratique(imb,lamb)
    errquad=norm2(im-resquad)
    print(lamb,errquad)
    if errquad<errbest:
        lambest=lamb
        errbest=errquad
print(lambest)
lambestquad=lambest
restquad=minimisation_quadratique(imb,lambestquad)
errquad=norm2(restquad-im)
viewimage(restquad,titre='BESTQUAD') # lambdaquad= 1.1220184543019658
#%% La partie variation totale
vk=np.arange(1.30,2,0.02)
errbest=norm2(imb)+10
for k in vk:
    lamb=10**k
    resvartot=vartotale_Chambolle(imb,lamb)
    errvartot=norm2(im-resvartot)
    print(lamb,errvartot)
    if errvartot<errbest:
        lambest=lamb
        errbest=errvartot
print(lambest)
lambestvartotale=lambest
restvartotale=vartotale_Chambolle(imb,lambestvartotale)
errvar=norm2(restvartotale-im)
viewimage(restvartotale,titre='BESTVARTOTALE')# lambdavartot=40.73802778041128
```

- Après avoir fixé une image bruitée par un bruit de 25. La meilleure valeur du paramètre lambda par la méthode de régularisation quadratique est égale à 1,220. Par contre, la meilleure valeur obtenue par la méthode de variation totale est égale à 40,738.
- On remarque que la valeur de errvartot est plus petite que la valeur de errquad. Ceci est clair lors de la visualisation des images car l'image résultante par la méthode de variation totale est bien meilleure.
- Pour la méthode de variation totale, Les bords ne sont pas détruits.