

# TP de méthodes variationnelles - IMA203

Mohamed Amir Belkacem

January 2024

## 1 Débruitage par Régularisation Quadratique

Le modèle d'observation est

$$v = u + b \quad (1)$$

où  $v$  est l'image observée (la donnée),  $u$  l'image parfaite et  $b$  le bruit.

On cherche à retrouver  $u$  comme minimiseur de l'énergie

$$E_1(u) = \|u - v\|^2 + \lambda \|\nabla u\|^2 \quad (2)$$

Le premier terme est simplement la norme au carré de la différence entre  $u$  et  $v$ . Le second terme est

$$\iint \|\nabla u(x, y)\|^2 dx dy \quad (3)$$

C'est-à-dire l'intégrale du carré du gradient de l'image en tout point.

1. Comment utiliser l'outil `resoud_quad_fourier` pour trouver le minimiseur de cette énergie (voir le programme minimisation quadratique) ?

Nous avons le problème:

$$\underset{u}{\operatorname{argmin}} \quad \|u - v\|^2 + \lambda \int \|\nabla u\|^2 \quad (4)$$

Nous pouvons réécrire:

$$E_1(u) = \|u - v\|^2 + \lambda \int \|\nabla u\|^2 = \|\delta * u - v\|^2 + \|\sqrt{\lambda} K_x * u - 0\|^2 + \|\sqrt{\lambda} K_y * u - 0\|^2 \quad (5)$$

Donc:

$$E_1(u) = \sum_{i=1}^N \|K_i * u - V_i\|^2 \quad \text{avec} \quad K = (\sqrt{\lambda} K_x, \sqrt{\lambda} K_y, \delta) \quad \text{et} \quad V = (0, 0, v) \quad (6)$$

D'après le théorème de Parseval en appliquant la transformée de Fourier:

$$\sum_{i=1}^N \sum_{\omega \in \mathcal{K}} |(\widehat{K_i * u - V_i})(\omega)|^2 = \sum_{i=1}^N \sum_{\omega \in \mathcal{K}} |\widehat{K_i}(\omega) \widehat{u}(\omega) - \widehat{V_i}(\omega)|^2 \quad (7)$$

Les deux sommes sont indépendantes, donc nous pouvons échanger  $f$  et  $i$ :

$$\sum_{\omega \in \mathcal{K}} \sum_{i=1}^N |\widehat{K_i}(\omega) \widehat{u}(\omega) - \widehat{V_i}(\omega)|^2 \quad (8)$$

En développant l'expression :

$$\sum_{\omega \in \mathcal{K}} \left[ |\widehat{u}(\omega)|^2 \left( \sum_{i=1}^N |\widehat{K_i}(\omega)|^2 \right) - 2 \operatorname{Re} \left( \overline{\widehat{u}(\omega)} \left( \sum_{i=1}^N \overline{\widehat{K_i}(\omega)} \widehat{V_i}(\omega) \right) \right) + \text{cte} \right] \quad (9)$$

Minimum est attendu pour  $\widehat{u}(\omega)$ :

$$\widehat{u}^*(\omega) = \frac{\sum_{i=1}^N \overline{\widehat{K_i}(\omega)} \widehat{V_i}(\omega)}{\sum_{i=1}^N |\widehat{K_i}(\omega)|^2} \quad (10)$$

La fonction `minimisation_quadratique` prend comme arguments  $v$  et  $\lambda$  pour construire  $K$  et  $V$ . Elle fait appel ainsi à la fonction `resoud_quad_fourier` qui prend comme arguments  $V$  et  $K$ , calcule  $\widehat{u}^*(\omega)$  puis effectue une transformée de Fourier inverse.

2. Décrire le résultat de ce débruitage lorsque  $\lambda$  est très grand ou très petit.

**Pour  $\lambda$  très petit :**

- Le terme de régularisation devient négligeable par rapport au terme d'attache aux données.
- La solution  $u$  tend à être très proche de l'image observée  $v$ .
- Le débruitage est faible car la régularisation n'influence pas significativement la solution.

**Pour  $\lambda$  très grand :**

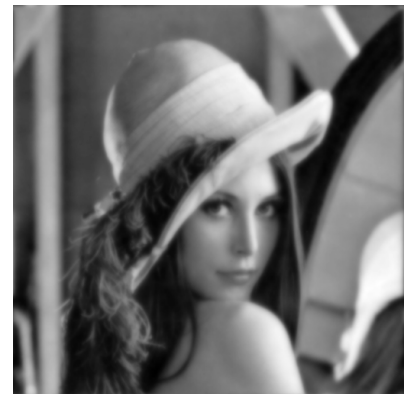
- Le terme de régularisation domine le terme d'attache aux données.
- La solution  $u$  est fortement influencée par la régularisation du gradient.
- La solution tend à être lisse, en minimisant le changement d'intensité entre pixels, mais cela peut entraîner une perte de détails fins par rapport à l'image observée  $v$ .



(a)  $\lambda$  égale à 0.1



(b)  $\lambda$  égale à 1



(c)  $\lambda$  égale à 10

Figure 1: Les résultats du débruitage de l'image Lena par régularisation quadratique sont présentés pour des valeurs de  $\lambda$  égales à 0.1, 1 et 10.

3. Après avoir ajouté un bruit d'écart type  $\sigma = 5$  à l'image de Lena, trouver (par dichotomie) le paramètre  $\lambda$  pour lequel  $\|\tilde{u} - v\|^2 \approx \|u - v\|^2$ . C'est-à-dire le paramètre pour lequel l'image reconstruite  $\tilde{u}$  est à la même distance de l'image parfaite  $u$  que ne l'est l'image dégradée.

En utilisant la méthode de dichotomie, nous avons débuté avec un intervalle initial de  $\lambda$  suffisamment étendu. À chaque étape, nous calculons le point médian  $\frac{\lambda_{\min} + \lambda_{\max}}{2}$  et remplaçons l'une des valeurs. Grâce à l'algorithme décrit par 2, nous avons abouti à  $\lambda \approx 0.33$ .

```
m = 0
M = 10
eps = 1e-3
dist = 1
while(np.abs(dist)>eps and M>=m):
    mid = (m+M)/2
    restq=minimisation_quadratique(imb,mid)
    dist = norm2(imb-restq)- norm2(imb)
    #print(dist)
    if(dist > eps):
        M = mid
    if(dist < -eps):
        m = mid
lamb_opt = mid
print(lamb_opt)
```

0.3315770626068115

(a) Le code pour le calcul par dichotomie du paramètre  $\lambda$



(b) l'image résultante avec  $\lambda \approx 0.33$

Figure 2: Le code pour le calcul dichotomique de  $\lambda$  et l'image correspondante avec la valeur trouvée.

4. Écrire un algorithme pour trouver le paramètre  $\lambda$  tel que  $\|\tilde{u} - u\|^2$  soit minimale, (dans le cadre de ce TP on connaît l'image parfaite  $u$ , on généralise on ne la connaît pas).  
Commentaires ?

```

%% COMPARAISON des methodes
# vous pouvez vous inspirer de ce qui suit pour trouver les meilleurs
# parametres de regularisation

errq=[]
errvt=[]
erreur=[]
vk=np.arange(-2,0,0.01)
for k in vk:
    #print (k)
    restq=minimisation_quadratique(imb,10**(k))
    erreur.append(norm2(im-restq))
plt.plot(10**vk,erreur)

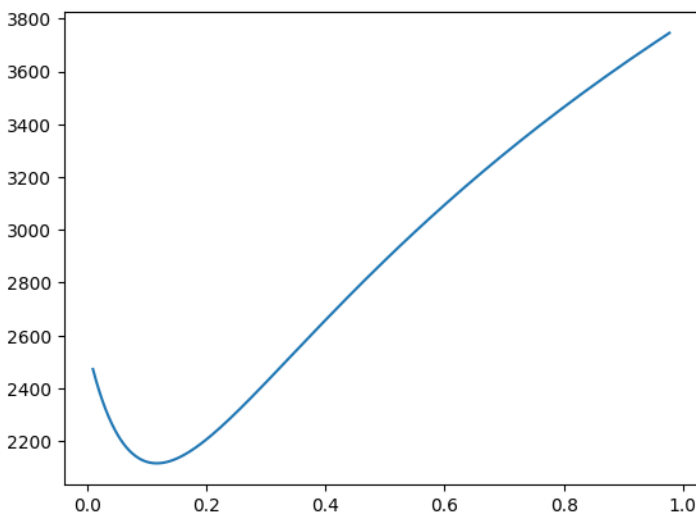
vkmin = vk[np.argmin(erreur)]
lambda_min = 10**vkmin
print(lambda_min)

```

0.11748975549395321

Figure 3: L'algorithme pour chercher le  $\lambda$  optimale

L'algorithme consiste à calculer l'image débruitée pour un intervalle de  $\lambda$  et à évaluer la norme de la différence avec l'image parfaite. Ensuite, on recherche la valeur minimale de cette norme.



(a) Courbe de la norme  $\|\tilde{u} - u\|^2$  en fonction de  $\lambda$



(b) L'image résultante avec  $\lambda \approx 0.117$  à 100 itérations

Figure 4: Résultat de l'algorithme de recherche de  $\lambda$  optimale

Sur le graphique, la valeur optimale de  $\lambda$  est de 0.117, minimisant la distance. En augmentant  $\lambda$ , le résultat se rapproche de l'image d'origine grâce à une régularisation accrue. Cela conduit à une convergence des statistiques de l'image traitée vers celles d'une image naturelle, produisant ainsi une version débruitée. Cependant, au-delà d'un certain point, la régularisation altère l'essence même de l'image, entraînant une perte de corrélation avec les données originales et générant une image floue, notablement différente de l'original.

## 2 Débruitage par par variation totale

### 2.1 Descente de gradient

1. Atteignez-vous toujours le même minimum d'énergie? (le programme renvoie l'évolution de l'énergie)

En utilisant le programme `minimise_TV_gradient` avec différentes valeurs du pas de descente, la sortie varie effectivement. Cette variabilité dans les résultats indique une sensibilité du processus de descente de gradient à

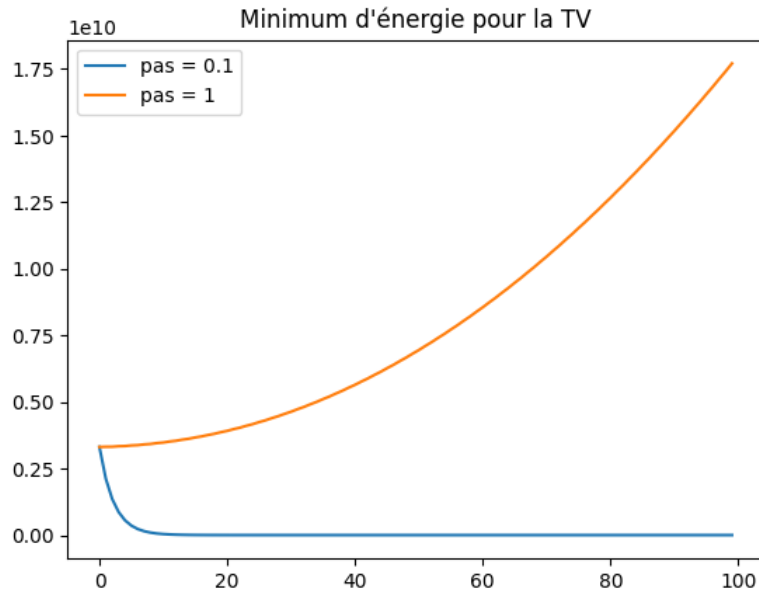


Figure 5: Minimum d'énergie trouvé par pas de 0.1 et 1

la sélection du pas, mettant en évidence les défis numériques associés à cette approche pour la minimisation de la variation totale dans le contexte du débruitage. Ainsi, il est confirmé que l'atteinte du même minimum d'énergie n'est pas garantie avec cette méthode, renforçant l'idée de la limitation de la descente de gradient à pas constant dans ce scénario.

## 2.2 Projection Chambolle

1. Utilisez ce programme et que constatez-vous quant à la vitesse de cet algorithme et sa précision (minimisation effective de  $E_2$ ) par rapport à la descente de gradient.

```
from time import time
t1 = time()
(u_grad, energ_grad)=minimise_TV_gradient(imb,10,0.1,100)
d_grad = time()-t1
E2_grad=E2_nonperiodique(u_grad,imb,10)
print("L'énergie E2 par la TV = ",E2_grad)
print("Le temps d'execution = ", d_grad)

L'énergie E2 par la TV = 26417909.318613775
Le temps d'execution = 1.1569116115570068

t2=time()
u_ch=var_totale_chambolle(imb,10,itmax=100)
d_ch=time()-t2
E2_ch=E2_nonperiodique(u_ch,imb,10)
print("L'énergie E2 par la projection chambolle = ",E2_ch)
print("Le temps d'execution = ", d_ch)

L'énergie E2 par la projection chambolle = 23894199.655030467
Le temps d'execution = 0.8672430515289307

print("Différence de temps d'exécution (grad- chambolle) = ", d_grad - d_ch)
print("Rapport d'énergie (grad/chambolle) = ", E2_grad/E2_ch)

Différence de temps d'exécution (grad- chambolle) = 0.28966856002807617
Rapport d'énergie (grad/chambolle) = 1.1056201797933831
```



(a) Le code pour comparer la descente de gradient et la projection Chambolle

(b) l'image résultante avec la projection Chambolle pour  $\lambda = 10$  à 100 itérations

Figure 6: Comparaison entre la descente de gradient et la projection Chambolle

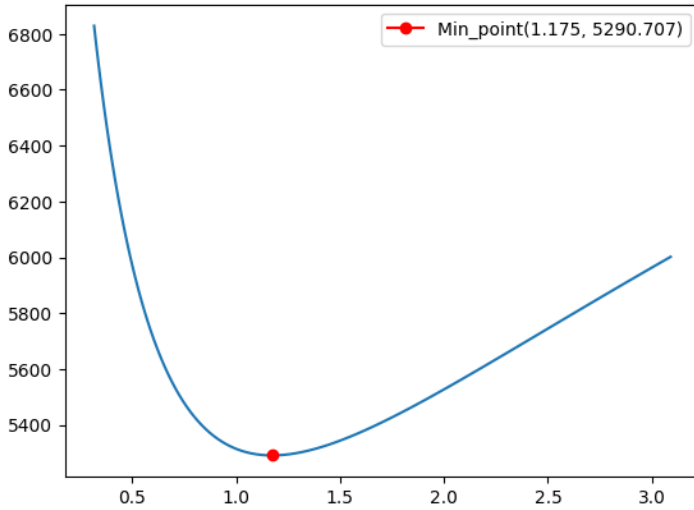
Il est notable que la minimisation par descente de gradient, avec un pas de 0.1 et 100 itérations, prend plus de temps que le processus effectué avec l'algorithme de Chambolle. (réduction de temps d'exécution par  $\approx 0.3$ ). Par ailleurs, l'algorithme de Chambolle démontre une amélioration de l'énergie, manifestée par une valeur plus réduite. (Rapport d'énergie grad/chambolle  $\approx 1.1$ ).

### 3 Comparaison

Après avoir fixé une image bruitée avec un bruit de 25, trouvez le meilleur paramètre  $\lambda$  pour chacune des deux méthodes (TV et quadratique) et comparez qualitativement les résultats obtenus par les deux méthodes pour le débruitage.

#### 3.1 Calcul des meilleurs lambdas

Pour la régularisation quadratique



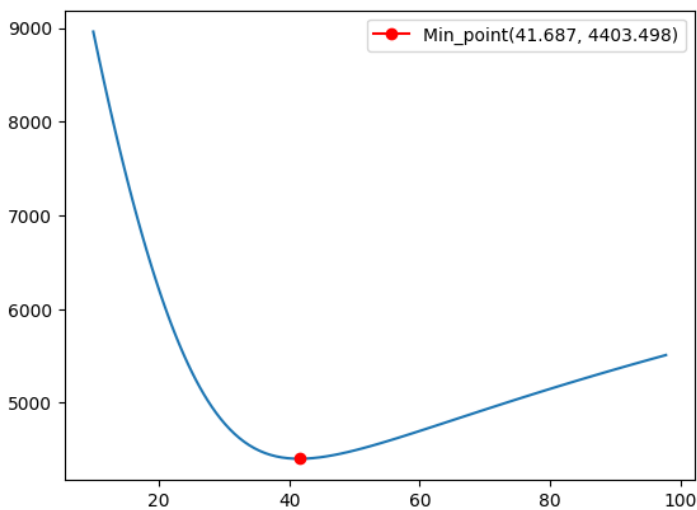
(a) La courbe de l'énergie en fonction de  $\lambda$



(b) l'image résultante avec la regularisation quadratique pour  $\lambda = 1.175$

Figure 7: Résultat de l'algorithme de recherche de  $\lambda$  optimale pour la régularisation quadratique

Pour la variation totale



(a) La courbe de l'énergie en fonction de  $\lambda$



(b) l'image résultante avec la regularisation quadratique pour  $\lambda = 41.687$

Figure 8: Résultat de l'algorithme de recherche de  $\lambda$  optimale pour la variation totale

## 3.2 Calcule du temps d'exécution

Pour la régularisation quadratique Pour la variation totale

```
t4 = time()
restq2 = minimisation_quadratique(imb2,lambda_min2)
d_quad = time()-t4
print("Le temps d'execution pour la régularisation quadratique = ", d_quad)
```

Le temps d'execution pour la régularisation quadratique = 0.32499194145202637

Figure 9: Temps d'exécution pour la régularisation quadratique

```
t5 = time()
restq3 = vartotale_Chambolle(imb2,lambda_min3)
d_ch2 = time()-t5
print("Le temps d'execution pour la variation totale = ", d_ch2)
```

Le temps d'execution pour la variation totale = 1.251842737197876

Figure 10: Temps d'exécution pour la variation totale

## 3.3 Interpretation

La minimisation de l'énergie  $E_2$  (utilisant la méthode de variation totale) présente une amélioration significative de l'erreur, et visuellement, l'image semble de meilleure qualité que celle obtenue avec la régularisation quadratique. En outre, au niveau des contours, on observe que ceux-ci se rapprochent davantage de ceux de l'image initiale lorsqu'on utilise la méthode de variation totale. Bien que cette méthode nécessite légèrement plus de temps de calcul, elle demeure très raisonnable. Ainsi, la méthode par variation totale se révèle être la meilleure option.