

## TRAVAUX PRATIQUES N°2

### Débruitage d'images

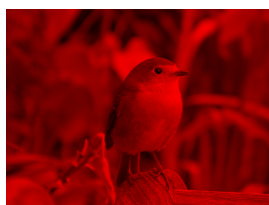
**Objectif :** Dans ce TP, on étend le concept du débruitage de signal aux images en niveaux de gris, puis on compare différentes fonctionnelles pour résoudre ce problème dans le cas d'un bruit blanc gaussien.

## 1 Introduction à l'imagerie numérique

Une image couleur peut être représentée par un élément  $u$  de  $\mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^c$ , où  $m \times n$  correspond au nombre de pixels et  $c$  le nombre de canaux couleurs. Traditionnellement, on en utilise généralement  $c = 3$  pour les images usuelles (c'est-à-dire les photographies grand public, à comparer avec les images obtenues par d'autres procédés comme la radiographie médicale). Ces trois canaux correspondent aux canaux R (rouge), G (vert) et B (bleu) (voir figure 1). Une autre classe d'images est celle des images en niveaux de gris, représentée par une matrice de  $\mathbb{R}^m \times \mathbb{R}^n$  (auquel cas  $c = 1$ ).



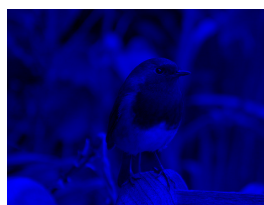
(a)



(b)



(c)



(d)

FIGURE 1 – (a) Exemple d'images en couleur. (b) Canal rouge. (b) Canal vert. (c) Canal bleu. Le canal bleu apparaît très sombre car cette composante est peu présente dans l'image (a). On voit *a contrario* qu'au niveau de la face du rouge-gorge, les plumes sont plus claires dans le canal rouge que dans le canal vert, puisqu'elles sont orangées, tandis que le fond, plutôt vert, apparaît plus clair dans le canal vert que dans le canal rouge.

Les valeurs que peuvent prendre les coefficients de  $u$  constituent la *dynamique* de l'image. Pour les images codées en 8 bits (par exemple les images JPEG, PNG), il s'agit de valeurs entières comprises entre 0 et 255 (il y a donc 256 valeurs possibles). Pour une image en niveaux de gris, une valeur proche de 0 correspond à une couleur sombre (proche du noir)

et une valeur proche de 255 correspond à une couleur claire (proche du blanc). Fondamentalement, ces valeurs doivent être rapprochées du nombre de photons mesuré par le capteur associé au pixel considéré. Une autre convention consiste alors à normaliser les valeurs de  $u$ , de sorte qu'elles appartiennent à l'intervalle (continu)  $[0; 1]$ . Pour une image codée en 8 bits, il suffit de diviser  $u$  par 255.

Dans ce TP, on va se concentrer sur les images en niveaux de gris, pour simplifier les calculs. Il existe beaucoup de manière de convertir une image couleur en niveaux de gris. La plus simple consiste à moyenner les trois canaux couleur.

La plupart des langages de programmation permet d'importer et d'afficher les images. En Matlab par exemple, la fonction `imread` permet d'importer une image et de la convertir en un élément de  $\mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^3$ , tandis que la fonction `imshow` ou `imagesc` permet de les afficher.

## 2 Débruitage d'une image numérique

Entre l'acquisition (prise de vue) à l'aide d'un appareil photo d'une scène, et le fichier numérique contenant l'image de cette scène, il y a une succession de phénomène qui altère l'image obtenue. On parle de *bruit*. Ce phénomène est particulièrement visible lorsque l'on essaie de prendre une photo avec une très faible luminosité. En réalité, le bruit est toujours présent, mais dans le cas d'une scène sombre, le signal (la luminosité des points de la scène que l'on cherche à capturer) est faible, du même ordre de grandeur que le bruit, de sorte que ce dernier est plus visible.

L'opération qui cherche à supprimer le bruit d'une image, ou à restaurer la qualité initiale d'une image, est appelée *débruitage*. Pour la réaliser, il faut commencer par *modéliser* le phénomène du bruit. Un modèle simple et populaire est celui du *bruit blanc gaussien* : on suppose que l'image capturée  $v$  est la somme d'une image idéale  $u$  et d'un bruit blanc gaussien  $b$  :

$$v = u + b$$

Plus précisément, le bruit ne dépend pas de la couleur du point considéré, est indépendant en chaque point de l'image, et suit la même loi gaussienne.

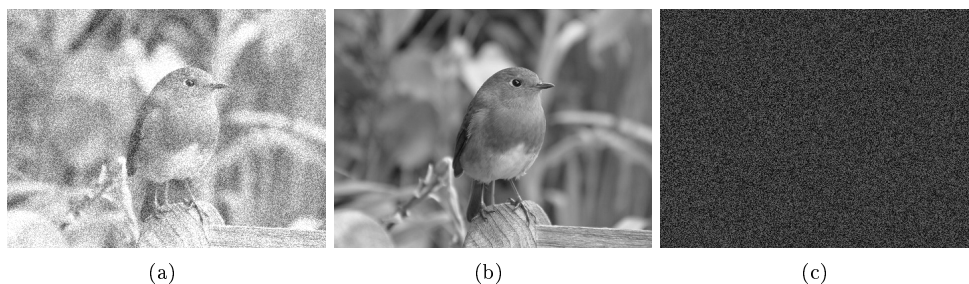


FIGURE 2 – (a) Image bruitée  $v$ . (b) Image idéale  $u$ . (c) Bruit  $b$ .

Pour plus de précisions sur le problème du débruitage et les méthodes variationnelles, on pourra consulter le [TP n°1 : Débruitage d'un signal 1D](#).

Pour débruiter une image en niveaux de gris, on considère dans ce TP les méthodes variationnelles. On rappelle que ces méthodes consistent à définir une fonctionnelle  $J$  permettant de mesurer la distance d'une image au modèle de formation que l'on a construit. On transforme alors le problème de débruitage en problème d'optimisation. Si  $v$  (l'image bruitée) est une image en niveaux de gris de taille  $m \times n$ , et si le bruit est un bruit additif gaussien, alors on peut considérer une classe de fonctionnelles de la forme

$$J(x) = \frac{\lambda}{2} \|v - x\|_2^2 + R(x)$$

avec  $x \in \mathbb{R}^{m \times n}$ ,  $R : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^+$  et  $\lambda > 0$ . Le premier terme est appelé *terme d'attache aux données*, ou *terme de fidélité*. Ce terme force la reconstruction (idéalement, l'image  $u$ ) à rester proche de l'image bruitée. Il y a donc de multiples façons de le définir. Ici, on choisit la distance euclidienne au carré à l'image bruitée  $v$  (erreur quadratique moyenne), car il a été démontré que ce choix est adapté au modèle choisi pour le bruit (voir encart plus bas). Le terme  $R$ , appelé *terme de régularisation*, traduit le modèle choisi pour l'image. On verra dans ce TP plusieurs choix possibles, qui sont de la forme

$$R(x) = r(\nabla^h x)$$

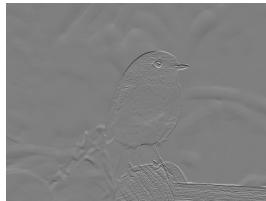
où  $\nabla^h x = (d_x x, d_y x)$  est la discrétisation du gradient, défini à l'aide de différences finies :

$$\forall (i, j) \in \llbracket 1; m \rrbracket \times \llbracket 1; n \rrbracket, \quad (\delta_x^h x)_{(i, j)} = \begin{cases} x_{(i+1, j)} - x_{(i, j)} & \text{si } i \neq m \\ 0 & \text{sinon} \end{cases}$$

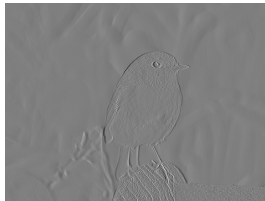
$$\text{et} \quad \forall (i, j) \in \llbracket 1; m \rrbracket \times \llbracket 1; n \rrbracket, \quad (\delta_y^h x)_{(i, j)} = \begin{cases} x_{(i, j+1)} - x_{(i, j)} & \text{si } j \neq n \\ 0 & \text{sinon} \end{cases}$$



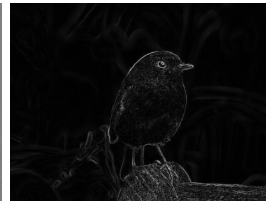
(a)



(b)



(c)



(d)

FIGURE 3 – (a) Image originale en niveaux de gris. (b) Dérivée verticale  $\delta_x^h u$ . (c) Dérivée horizontale  $\delta_y^h u$ . (d) Norme du gradient  $\|\nabla^h u\|$ . Pour les deux premières images (b) et (c), les valeurs sont comprises entre  $-0.5$  (noir) et  $0.5$  (blanc). Pour la dernière image, les valeurs sont comprises (après normalisation) entre  $0$  (noir) et  $1$  (blanc). On voit que les valeurs les plus grandes correspondent à des pixels situés sur les contours des objets dans l'image. Cette propriété justifie que le gradient de l'image soit souvent utilisé comme un *détecteur de contours*.

On peut rapprocher ces deux opérateurs de l'opérateur  $D$  introduit dans le TP n°1 : Débruitage de signaux 1D. Il s'agit des approximations de la dérivée verticale et de la dérivée horizontale, respectivement, d'une fonction  $\tilde{u}$  définie sur le plan, telle que

$$\forall (i, j) \in \llbracket 1; m \rrbracket \times \llbracket 1; n \rrbracket, \quad \tilde{u}(i, j) = u_{i,j}$$

L'idée sous-jacente est que le gradient d'une image "ordinaire" (photographies par exemple) possède des propriétés caractéristiques. Enfin, le paramètre  $\lambda$  permet de pondérer l'importance relative de ces deux termes.

Plus  $\lambda$  est grand, plus le terme d'attache aux données est prépondérant ; autrement dit, plus  $\lambda$  est grand, plus le minimiseur de  $J$  sera proche des données initiales bruitées  $v$ . C'est un bon choix lorsque l'image  $v$  est peu bruitée. *A contrario*, lorsque l'image  $v$  est très bruitée, il faut s'en éloigner pour obtenir un résultat satisfaisant, on doit donc accentuer la régularisation. On obtient un tel résultat en choisissant une valeur faible pour  $\lambda$ . Le réglage de ce paramètre est donc indispensable pour obtenir le meilleur résultat possible. En théorie, sa valeur optimale est liée au niveau de bruit dans l'image.

### Exercice 1 – Divergence d'une image

On note  $\operatorname{div}^h$  l'opposé de l'opérateur adjoint à l'opérateur  $\nabla^h$  :

$$\operatorname{div}^h = -(\nabla^h)^*$$

Montrer que, pour tout  $w = (w^x, w^y) \in \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n}$ ,

$$\operatorname{div}^h w = -(\delta_x)^* w^x - (\delta_y)^* w^y$$

avec

$$\forall (i, j) \in \llbracket 1; m \rrbracket \times \llbracket 1; n \rrbracket, \quad ((\delta_x)^* w^x)_{(i,j)} = \begin{cases} -w_{(1,j)}^x & \text{si } i = 1 \\ w_{(i-1,j)}^x - w_{(i,j)}^x & \text{si } 1 < i < m \\ w_{(m-1,j)}^x & \text{sinon} \end{cases}$$

et

$$\forall (i, j) \in \llbracket 1; m \rrbracket \times \llbracket 1; n \rrbracket, \quad ((\delta_y)^* w^y)_{(i,j)} = \begin{cases} -w_{(i,1)}^y & \text{si } j = 1 \\ w_{(i,j-1)}^y - w_{(i,j)}^y & \text{si } 1 < j < n \\ w_{(i,n-1)}^y & \text{sinon} \end{cases}$$

En déduire que

$$|||\nabla^h||| \leq 2\sqrt{2}$$

Pour vérifier que les opérateurs  $\nabla^h$  et  $\operatorname{div}^h$  sont bien implémentés, il faut vérifier que

$$\langle u, w \rangle, \quad \langle \nabla^h u, w \rangle = -\langle u, \operatorname{div}^h w \rangle$$

Numériquement, il suffit de le vérifier pour deux éléments  $u$  et  $w$  tirés aléatoirement.

### 3 Régularisation de TIKHONOV

Nous allons commencer par un modèle simple, que l'on a déjà vu dans le TP n°1 : **Débruitage de signaux 1D**. Dans ce modèle, on suppose que la différence de valeurs entre deux pixels voisins est faible dans l'image que l'on souhaite restaurer ( $u$ ). Autrement dit, que les variations d'intensité (de niveaux de gris) dans une image sont douces. Dans ce cas, en posant

$$r(\nabla^h x) = \frac{1}{2} \|\nabla^h x\|_2^2$$

on définit un terme de régularisation qui prendra une valeur d'autant plus élevée que l'image  $x$  présente des variations brutales d'intensité (des paires de pixels voisins ayant des valeurs très différentes). Ce terme mesure donc bien une distance au modèle choisi. Dans ce modèle, les images  $v$  telles que  $r(\nabla^h x)$  soit le plus faible sont les images constantes (qui ne présentent donc aucune variation d'intensité). On se retrouve donc à considérer le problème d'optimisation suivant

$$\min_{x \in \mathbb{R}^{m \times n}} J_1(x) = \frac{\lambda}{2} \|v - x\|_2^2 + \frac{1}{2} \|\nabla^h x\|_2^2 \quad (\mathcal{P}_1)$$

#### Exercice 2 – Propriétés de $J_1$

- (a) Montrer que  $J_1$  est différentiable et que, pour tout  $x \in \mathbb{R}^{m \times n}$

$$\nabla J_1(x) = \lambda(x - v) - \operatorname{div}^h(\delta_x^h, \delta_y^h x)$$

- (b) Montrer que  $J_1$  est convexe et fortement convexe de module  $\lambda$ .  
 (c) Montrer que  $\nabla J_1$  est lipschitzien, de constante de LIPSCHITZ

$$L_1 = \lambda + 8$$

Pour les expériences numériques, on va importer une image visuellement satisfaisante (n'importe quelle photographie sans défaut visible), puis ajouter un bruit blanc gaussien. Attention : il est possible que l'image importée comporte déjà du bruit, de sorte que la considérer comme le signal idéal  $u$  n'est pas très rigoureux. Cependant, on va ici rajouter un bruit très important (visible), de sorte que cette hypothèse restera raisonnable. Il faut également veiller à importer des images de taille suffisamment petite, pour éviter de manipuler des objets trop grands.

#### Exercice 3 – Méthode du gradient à pas fixe

Implémenter la méthode du gradient à pas fixe pour résoudre le problème  $(\mathcal{P}_1)$ . On choisira  $\lambda = 0.1$  et un nombre d'itérations de 100. On initialisera avec  $v_0 = g$ . On pensera, dans tout ce TP, à afficher l'évolution de  $J_i(x_k)$  et  $\|\nabla J_i(x_k)\|$ .

On présente dans la figure 4 le résultat obtenu à l'aide du modèle de la régularisation de TIKHONOV. On voit que l'image reconstruite ne semble plus bruitée (elle a perdu le grain que l'on observait dans l'image bruitée). Cependant, elle est loin du résultat espérer, à savoir l'image  $u$  ! En effet, cette image est floue, alors que l'image initiale est nette. Cela est dû au choix de la régularisation : à cause du choix de la norme quadratique pour  $r$ , la solution du problème  $(\mathcal{P}_1)$  présente un gradient de norme faible, c'est-à-dire des variations douces. Or, des variations douces se traduisent par un effet de flou.

Ici, ce n'est pas la résolution du problème qui est en cause, mais bien le choix du problème d'optimisation, et, donc, du modèle sous-jacent. On verra dans la prochaine section un modèle plus satisfaisant, mais au prix d'un problème d'optimisation plus difficile à résoudre !



FIGURE 4 – (a) Image bruitée  $v$ . (b) Image reconstruite par la régularisation de TIKHONOV.

## 4 Modèle de RUDIN–OSHER–FATEMI

Ce modèle testé dans la section précédente, simpliste, est évidemment peu adapté aux images courantes, dans lesquels on peut observer des transitions brusques d'intensité, survenant en particulier sur les contours des objets (par exemple un objet sombre sur un fond clair), ou encore dans des motifs (exemple : un échiquier). On le verra de manière très claire dans les expériences numériques. Dans cette section, on introduit un modèle plus complexe, mais plus proche de la réalité. En effet, les images dite naturelles présentent des caractéristiques intéressantes : pour simplifier, elles comportent des zones avec des variations très douces (voire pas de variations du tout, autrement dit des aplats) délimitées par des bords francs. Il a alors été démontré qu'une meilleure régularisation consiste à remplacer la norme quadratique par la norme euclidienne :

$$r(\nabla^h x) = \|\nabla^h x\|$$

Ce choix est à rapprocher d'un concept plus vaste, celui de la *variation totale* (voir encart plus bas). C'est pourquoi on parle de régularisation TV (pour *Total Variation*).

Il est en réalité plus adapté de représenter une image, non comme une matrice  $u$  (c'est-à-dire une grille de pixels), mais une fonction  $\tilde{u}$  (c'est-à-dire définie sur un domaine continu, un rectangle  $\Omega$  par exemple). Or, les images naturelles présentent une variation totale

$$\text{TV}(\tilde{u}) = \int_{\Omega} |D\tilde{u}|$$

faible. On notera que, lorsque  $\tilde{u}$  est différentiable, sa variation totale vaut la norme de son gradient. La théorie mathématique générale sur cette quantité est celle de l'espace BV (*Bounded Variations*).

On est donc cette fois amené à considérer le problème d'optimisation suivant :

$$\min_{x \in \mathbb{R}^{m \times n}} J_2(x) = \frac{\lambda}{2} \|v - x\|_2^2 + \|\nabla^h x\| \quad (\mathcal{P}_2)$$

La fonction objectif  $J_2$  n'est pas différentiable cette fois-ci, à cause du second terme. En particulier, il n'est plus possible d'utiliser la méthode du gradient. Dans ce TP, on va tester deux approches différentes. Dans la première approche, on va modifier le problème  $(\mathcal{P}_2)$  pour en proposer une variante approchée, mais différentiable, de sorte de pouvoir appliquer la méthode du gradient. Malheureusement, une telle manipulation ne préserve pas les minimiseurs ; autrement dit, la solution que l'on obtiendra n'est pas assurée d'être (proche) de la solution du problème initial. Dans la seconde approche, on choisit de résoudre le problème initial  $(\mathcal{P}_2)$  exactement ; on verra qu'il faut alors considérer un algorithme plus compliqué.

Une première idée est donc de remplacer le terme non différentiable par une approximation différentiable, de gradient lipschitzien. Pour cela, on commence par s'intéresser à la fonction suivante :

$$\varphi_\alpha : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R}^+ \\ t & \mapsto |t| - \alpha \ln \left( 1 + \frac{|t|}{\alpha} \right) \end{cases} \quad \text{avec } \alpha > 0.$$

#### Exercice 4 – Propriétés de la fonction $\varphi_\alpha$

Soit  $\alpha > 0$ .

(a) Démontrer que  $\varphi_\alpha$  est de classe  $\mathcal{C}^2$  et que

$$\varphi'_\alpha : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R}^+ \\ t & \mapsto \frac{t}{\alpha + |t|} \end{cases} \quad \text{et} \quad \varphi''_\alpha : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R}^+ \\ t & \mapsto \frac{\alpha}{(\alpha + |t|)^2} \end{cases}$$

(b) En déduire que  $\varphi'_\alpha$  est lipschitzien, de constante de LIPSCHITZ

$$L \leq \frac{1}{\alpha}$$

(c) Démontrer que  $\varphi_\alpha$  est convexe.

La fonction  $\varphi_\alpha$  approche de manière différentiable la valeur absolue, l'approximation étant d'autant meilleure que  $\alpha$  est petit :

#### Exercice 5 – Approximation de la valeur absolue

Afficher dans un même graphe la courbe représentative de la valeur absolue et celle de  $\varphi_\alpha$  pour  $\alpha \in \{0.5, 1, 2\}$ , sur  $[-10; 10]$ .

Dans ce cas, on peut remplacer le second terme de  $J_2$  par une approximation de la norme, et ainsi considérer le problème

$$\min_{x \in \mathbb{R}^{m \times n}} \tilde{J}_2(x) = \frac{\lambda}{2} \|v - x\|_2^2 + \sum_{i=1}^m \sum_{j=1}^n \varphi_\alpha((\delta_x^h)_{i,j}) + \varphi_\alpha((\delta_y^h)_{i,j}) \quad (\tilde{\mathcal{P}}_2)$$

**Exercice 6 – Propriétés de la fonction  $\tilde{J}_2$** 

Soit  $\alpha > 0$ .

- (a) Démontrer que  $\tilde{J}_2$  est fortement convexe.
- (b) Démontrer que  $\tilde{J}_2$  est différentiable et que, pour tout  $x \in \mathbb{R}^{m \times n}$ ,

$$\nabla \tilde{J}_2(x) = \lambda(x - v) - \operatorname{div}^h((\varphi'_\alpha((\delta_x^h x_{i,j}))), (\varphi'_\alpha((\delta_y^h x_{i,j}))))$$

- (c) Démontrer que  $\nabla \tilde{J}_2$  est lipschitzien et estimer une constante de LIPSCHITZ.

**Exercice 7 – Méthode du gradient pour le problème  $(\tilde{\mathcal{P}}_2)$** 

Implémenter la méthode du gradient à pas fixe pour résoudre le problème  $(\tilde{\mathcal{P}}_2)$ . On choisira  $\lambda = 4$ , un nombre total d'itérations de 100 et  $\alpha \in \{0.01, 0.1\}$ . On initialisera avec  $x_0 = v$ .

Dans la figure 5, on compare les résultats obtenus pour les deux valeurs de  $\alpha$  testés. On voit que, lorsque  $\alpha$  est petit (donc l'approximation de la valeur absolue est meilleure), le résultat est plus satisfaisant. On peut toutefois vérifier que la convergence (en termes du critère) est plus lente. Cela est dû au fait que le pas de temps est plus petit (car il croît avec  $\alpha$ ). Autrement dit, une meilleure approximation nécessite un temps de calcul plus long.



FIGURE 5 – (a) Image reconstruite avec  $\alpha = 0.01$ . (b) Image reconstruite avec  $\alpha = 0.1$ .

On peut choisir d'autres manières d'approcher la valeur absolue de manière régulière.

**Exercice 8 – Autre approximation de la valeur absolue**

Pour les fonctions suivantes, justifier qu'il s'agit d'une approximation différentiable de la valeur absolue. Discuter des choix possibles pour  $\alpha > 0$  et des propriétés théoriques de  $\varphi_\alpha$ .

(a)  $\varphi_\alpha : t \mapsto \sqrt{t^2 + \alpha}$

(b)  $\varphi_\alpha : t \mapsto \begin{cases} t^2/(2\alpha) & \text{si } |t| \leq \alpha \\ |t| - \alpha/2 & \text{sinon} \end{cases}$

On revient à présent au problème initial  $(\mathcal{P}_2)$ .



**Exercice 9 – Dualité**

- (a) Démontrer que  $J_2$  est fortement convexe.  
 (b) Démontrer que le terme de fidélité  $g : x \mapsto \lambda \|v - x\|^2/2$  est simple et que

$$\text{prox}_{\tau g}(x^0) = \frac{x^0 + \tau \lambda v}{1 + \tau \lambda}$$

- (c) Démontrer que  $r : w \mapsto \|w\|$  est simple et que

$$\text{prox}_{\sigma r^*}(y^0) = \text{proj}_{\mathcal{B}(0,1)}(y^0)$$

- (d) On pose  $\mathcal{L}(x; y) = g(x) + \langle \nabla^h x, y \rangle - \chi_{\mathcal{B}(0,1)}(y)$

Montrer que

$$\sup_{y \in \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n}} \mathcal{L}(x; y) = J_2(x)$$

puis que

$$\inf_{x \in \mathbb{R}^{m \times n}} \mathcal{L}(x; y) = -\frac{\text{div}^h y + \lambda v}{\lambda} - \chi_{\mathcal{B}(0,1)}(y)$$

Les itérations de l'algorithme de CHAMBOLLE–POCK s'écrivent pour résoudre  $(\mathcal{P}_2)$  donc

$$(x_0, y_0, \bar{x}_0) \quad \text{et} \quad \forall k \in \mathbb{N}, \quad \begin{cases} y_{k+1} = \text{proj}_{\mathcal{B}(0,1)}(y_k + \sigma \nabla^h \bar{x}_k) \\ x_{k+1} = \frac{x_k + \tau \lambda \text{div}^h(y_{k+1}) + \tau g}{1 + \tau \lambda} \\ \bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k) \end{cases}$$

On pose  $\mathcal{G}(x, y) = J_2(x) + \frac{\text{div}^h y + \lambda v}{\lambda} + \chi_{\mathcal{B}(0,1)}(y)$

**Exercice 10 – Algorithme de CHAMBOLLE–POCK pour le problème  $(\mathcal{P}_2)$** 

Implémenter la méthode de CHAMBOLLE–POCK pour résoudre le problème  $(\mathcal{P}_2)$ . On choisira  $\lambda = 4$  et un nombre total d'itérations de 100. On initialisera avec  $x_0 = \bar{x}_0 = v$  et  $y_0 = \nabla^h x_0$ . On affichera l'évolution de  $J_2(x_k)$  et  $\mathcal{G}(x_k, y_k)$ .